

NAVAL POSTGRADUATE SCHOOL

Monterey , California



THESIS

P157586

A RELATIONAL DATABASE MANAGEMENT SYSTEM
FOR A ROK ARMY INFANTRY DIVISION
WITH PROBABILISTIC INVENTORY CONTROL
MODEL

by

Taeyong Park

December 1987

Thesis Advisor

Y.K. Mortagy

Approved for public release; distribution is unlimited.

T239126

REPORT DOCUMENTATION PAGE

1a REPORT SECURITY CLASSIFICATION			1b RESTRICTIVE MARKINGS			
2a SECURITY CLASSIFICATION AUTHORITY UNCLASSIFIED			3 DISTRIBUTION / AVAILABILITY OF REPORT Approved for public release; Distribution is unlimited			
2b DECLASSIFICATION / DOWNGRADING SCHEDULE						
4. PERFORMING ORGANIZATION REPORT NUMBER(S)			5 MONITORING ORGANIZATION REPORT NUMBER(S)			
6a NAME OF PERFORMING ORGANIZATION Naval Postgraduate School		6b OFFICE SYMBOL (If applicable) 54		7a NAME OF MONITORING ORGANIZATION Naval Postgraduate School		
6c ADDRESS (City, State, and ZIP Code) Monterey, California 93943-5100			7b ADDRESS (City, State, and ZIP Code) Monterey, California 93943-5100			
8a NAME OF FUNDING / SPONSORING ORGANIZATION		8b OFFICE SYMBOL (If applicable)		9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER		
8c ADDRESS (City, State, and ZIP Code)			10 SOURCE OF FUNDING NUMBERS			
			PROGRAM ELEMENT NO	PROJECT NO	TASK NO	WORK UNIT ACCESSION NO
11 TITLE (Include Security Classification) A RELATIONAL DATABASE MANAGEMENT SYSTEM FOR A ROK ARMY INFANTRY DIVISION WITH PROBABILISTIC INVENTORY CONTROL MODEL						
12. PERSONAL AUTHOR(S) Park, Taeyong						
13a. TYPE OF REPORT Master's Thesis		13b. TIME COVERED FROM _____ TO _____		14 DATE OF REPORT (Year, Month, Day) 1987 December		15 PAGE COUNT 161
16 SUPPLEMENTARY NOTATION						
17 COSATI CODES			18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number)			
FIELD	GROUP	SUB-GROUP	Database Management System;			
			Database;			
			Selective Management System			
19 ABSTRACT (Continue on reverse if necessary and identify by block number) <p>The purpose of this thesis is to provide an inventory database management system for a ROK Army Infantry Division.</p> <p>Since, 1973, the ROK Army has significantly improved its capability in logistics management by using computer systems. However, the operational level of command (the division) has some difficulties in meeting higher command requirements because of the unavailability of computer hardware and software. As of last year, the computer hardware was installed at the division level. User friendly software development has become another requirement for effective use of this computer hardware. To help meet this objective, this thesis provides a database management system for the manager who works at the division logistics section, and for the Quartermaster battalion which is the major unit to handle material within the division.</p>						
20 DISTRIBUTION / AVAILABILITY OF ABSTRACT <input checked="" type="checkbox"/> UNCLASSIFIED/UNLIMITED <input type="checkbox"/> SAME AS RPT <input type="checkbox"/> DTIC USERS				21 ABSTRACT SECURITY CLASSIFICATION Unclassified		
22a. NAME OF RESPONSIBLE INDIVIDUAL Y. K. Mortagy			22b TELEPHONE (Include Area Code) (408) 646-2360		22c. OFFICE SYMBOL Code 54 My	

To meet the objective, the author of this thesis concentrated on writing user friendly interface programs within the current logistics management system.

By applying the proposed system, the Army can improve one of its logistics objective, i. e. automatic data processing. This can contribute to the logistics management system implementation by allowing the division to generate more accurate reports in less time, and to improve inventory management by shortening the administrative process.

Approved for public release; distribution is unlimited.

A Relational Database Management System
for a ROK Army Infantry Division
with Probabilistic Inventory Control Model

by

Taeyong Park
Major, Republic of Korea Army
B.S., Korea Military Academy, 1978

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN MANAGEMENT

from the

NAVAL POSTGRADUATE SCHOOL
December 1987

Thesis
259286
1

ABSTRACT

The purpose of this thesis is to provide an inventory database management system for a ROK Army infantry division.

Since 1973, the ROK Army has significantly improved its capability in logistics management by using computer systems. However, the operational level of command (the division) has some difficulties in meeting higher command requirements because of the unavailability computer hardware and software. As of last year, the computer hardware was installed at the division level. User friendly software development has become another requirement for effective use of this computer hardware. To help meet this objective, this thesis provides a database management system for the manager who works at the division logistics section, and for the Quartermaster battalion which is the major unit to handle material within the division.

To meet the objective, the author of this thesis concentrated on writing user friendly interface programs within the current logistics management system.

By applying the proposed system, the Army can improve one of its logistics objective, ie automatic data processing. This can contribute to the logistics management system implementation by allowing the division to generate more accurate reports in less time, and to improve inventory management by shortening the administrative process.

TABLE OF CONTENTS

I.	INTRODUCTION	13
A.	INTRODUCTION	13
B.	THESIS ORGANIZATION	14
II.	ARMY LOGISTICS STRUCTURE AND ADPS IN THE KOREA ARMY	15
A.	ARMY LOGISTICS STRUCTURE	15
1.	Overview	15
2.	The Army Logistics Unit Organization	15
3.	Management Issues	16
B.	SELECTIVE MANAGEMENT	17
1.	Overview	17
2.	The ABC Classification system	17
C.	THE EXISTING SYSTEM	19
1.	The Planning-Programming-Executing-Evaluating System (PPBEES)	19
2.	Resource flow in the infantry division	20
3.	Inventory operation and control in divisions	21
4.	Current documentation and report forms	26
D.	CURRENT USE OF COMPUTERS FOR LOGISTICS IN ROK ARMY	27
1.	Overview	27
2.	ADP Support to logistics	28
E.	SUMMARY	29
III.	RELATIONAL DATABASE MODELS	30
A.	INTRODUCTION	30
B.	WHAT IS DATABASE?	30
C.	RELATIONAL DATABASE MODEL	32
1.	Relation	32

2.	Keys	33
3.	Record relationships	33
4.	Database Management System	33
IV.	THE PROPOSED SYSTEM	35
A.	OVERVIEW	35
B.	MANUAL SYSTEM	35
C.	THE PROPOSED SYSTEM STRUCTURE	35
D.	THE PROPOSED SYSTEM FUNCTIONS	36
1.	Transaction tracking	36
2.	Report Generation	37
3.	Limited analysis	39
4.	Use friendly interface	40
E.	SYSTEM LIMITATIONS	41
1.	Demand forecasting	41
2.	Statistical analysis of transactions	41
3.	ASL request	41
4.	Fund Accounting	41
F.	ADVANTAGES OF THE SYSTEM	42
V.	CONCLUSIONS	43
A.	CONCLUSIONS	43
B.	FURTHER STUDY	43
	APPENDIX A: STATISTICAL CONSIDERATION	45
	APPENDIX B: DATA STRUCTURE	49
1.	OVERVIEW	49
2.	DATA FILES	49
a.	Master	49
b.	Property file	50
c.	ASL file	50
d.	Customer file	51
e.	Stockout file	51
f.	Batch file	51
g.	Normal file	51

3.	INDEX FILES	51
APPENDIX C: APPENDIX SYSTEM STRUCTURE		54
1.	OVERVIEW	54
2.	PMMAIN	54
3.	MANAGFL	54
4.	TRANSACT	56
a.	StockRD	56
b.	TRbatch and TRissue	56
c.	TRreqlst	58
d.	TRstock	58
e.	TURTOLC, TRTURNIN and TRTUNLST	59
f.	Query	59
5.	REPORTS	65
6.	ANALYSIS	69
APPENDIX D: USER'S MANUAL		70
1.	INTRODUCTION	70
a.	Requirements	70
b.	Organization	70
2.	GETTING STARTED	70
a.	Configuration	71
b.	Summary of the proposed system disk.	71
3.	WORKING WITH THE PROPOSED SYSTEM	73
a.	Routine transaction	73
b.	File management	76
c.	Reports generation	83
d.	Analysis	84
e.	Query	87
APPENDIX E: PROGRAM LISTING		89
1.	MAIN MENU	89
a.	PMAIN	89
b.	PMMENU	91
2.	ROUTINE TRANSACTION	93

a.	TRANSACTION MENU	93
b.	TRSTOCKRD	95
c.	TRBATCH	97
d.	TRISSUE	99
e.	TRREQLST	102
f.	TRSTOCK	105
g.	TRTURN-IN	110
h.	TRTUNLST	111
i.	TRCANCEL	113
j.	TRACKING	114
3.	REPORTS	116
a.	REPORTS MENU	116
b.	RPTSR	117
c.	RPOST	121
d.	RPEIS	125
e.	RPSOR	128
4.	ANALYSIS	132
a.	ANALYSIS STARTUP	132
b.	ANLEADT	133
c.	ANEOQ	135
d.	ANPROC	137
5.	MANAGEFL	139
a.	MANAGEFL MENU	139
b.	ADDASL	140
c.	ADDPROP	142
d.	ADDCUST	144
e.	EDITASL	144
f.	EDITBAT	145
g.	EDITCUST	146
h.	EDITMAST	147
i.	EDITPROP	148
j.	MANAGEQ	149

APPENDIX F: SCREEN FORMAT	151
1. TRANSACT	151
a. BATCH	151
b. MAST	151
c. TURNIN	152
d. CANCMASST	152
2. MANAGEFL	153
a. ASL	153
b. CUST	153
c. PROP	154
d. EDITBAT	154
e. EDITCUST	155
f. EDITASL	155
g. EDITPROP	156
h. EDITCUST	156
3. ANALYSIS	157
a. EOQ	157
LIST OF REFERENCES	158
INITIAL DISTRIBUTION LIST	159

LIST OF TABLES

1. BUDGET RATE OF MATERIALS	20
2. DEMAND FREQUENCY RATE	24
3. DATA STRUCTURE	50
4. DATA DICTIONARY	52
5. INDEX FILES	53

LIST OF FIGURES

2.1	ABC Classification	18
2.2	Division Supply Procedure	22
2.3	Army Logistics Computer System	27
3.1	View of database	31
3.2	Relation of property	32
3.3	Record relationship	34
4.1	Examples of transaction output	37
4.2	Main data flow diagram	38
4.3	Transaction data flow diagram	38
4.4	Report data flow diagram	39
4.5	The output from analysis process	40
C.1	Detailed system structure	55
C.2	Pmmain screen	56
C.3	ADDPROP module flowchart	57
C.4	Example of MANAGEQ module	58
C.5	Transaction menu screen	59
C.6	TRSTOKRD (Request to division) module flow chart	60
C.7	TRBATCH (batch process) module flow chart	61
C.8	TRSTOCK (Issue from LSC) flow chart	62
C.9	TURNTULC (Turn-in to LSC) module flowchart	63
C.10	Example of Query on transaction module	64
C.11	Report module menu screen	65
C.12	RPTSR (transaction status report) module flow chart	66
C.13	RPOST (Order shipping time report) module flow chart	67
C.14	RPSOR (Stockout report) module flow chart	68
C.15	Analysis module screen	69
D.1	Main menu(PMAIN)	73
D.2	Routine transaction menu screen	74

D.3	Request for issue from customer screen	75
D.4	Batch process examples	76
D.5	Request for turn-in from customer	77
D.6	Receive stock item from LSC	77
D.7	Cancellation of request screen	78
D.8	File management screen	78
D.9	Customer addition screen	79
D.10	Property list addition screen	79
D.11	ASL list addition screen	80
D.12	Customer record editing screen	81
D.13	ASL list record editing screen	81
D.14	Master file record editing screen	82
D.15	Batch file record editing screen	83
D.16	Reports generation menu screen	84
D.17	Transaction status report option menu	84
D.18	Examples of Transaction status reports	85
D.19	Analysis start-up screen	86
D.20	Variable entering for analysis	87
D.21	Transaction tracking	88
D.22	Query on stock number	88

I. INTRODUCTION

A. INTRODUCTION

This thesis proposes an inventory control database management system for use at the ROK Army division.

Supply functions are critical in achievement of economic operations of armed forces, and the management of defense resources. Republic of Korea (ROK) Army logisticians have developed better ways to build support process. One improvement is the recent computerization of Planning-Programming-Budgeting-Executing-Evaluating-System (PPBEES)¹ at high level command (above division).

However, implementation of PPBEES below the division level requires a great deal of manual paperwork, and much overtime for lower level managers.

Fortunately, as of last year, the Army divisions have acquired computer hardware. Still, some usage difficulties remain because of shortages of the computer-related personnel. Thus the development of well structured, user friendly computer systems is urgently needed.

This thesis will introduce a logistics database system which will reduce the amount of paperwork, decrease order processing response time and generate statistics which are now too cumbersome to compute. The system must be well structured and user friendly for easy implementation by novice users.

Therefore this thesis will deal with the application of computer based routine transactions, generation of reports, and analysis of transactions for the ROK Army division logistics management section.

The software developed for this thesis is microcomputer based because of price and availability to the Korea Army, especially at the division level and below. The software performs the supply distribution function, which is the most costly and labor consuming phase at the operational level.

It handles authorized storage list (ASL) as well as property items, to be defined at chapter II. The fixed order quantity with probabilistic demand model will be applied for inventory control.

¹PPBEES is a newly developed system for the national resource management in Korea. The objective is to accomplish the economic management of the Army by evaluation of managers with budget expended and the designated material readiness condition.

B. THESIS ORGANIZATION

The thesis is divided into the following sections: As the background of the thesis, the second chapter covers the current ROK Army logistics structure, supply functions, and current computers in the Korean Army.

The third chapter describes the structure and the capability of the proposed system and its benefits.

Future research needs will be stated in the concluding chapter. Appendices include statistical considerations for inventory control measures in Appendix A, data structure of the proposed system in Appendix B, the proposed system structure in Appendix C. The user's manual is provided in Appendix D, and program listings and menu screen formats are in Appendix E and F.

II. ARMY LOGISTICS STRUCTURE AND ADPS IN THE KOREA ARMY

A. ARMY LOGISTICS STRUCTURE

1. Overview

The Korea Army is the largest of the three armed forces and is responsible for providing general-purpose forces to meet any threat to the Korean national security. In order for the Army to carry out its mission, it has to insure an uninterrupted supply of weapons, equipment, supplies and other items to the combat forces.

The Army logistic unit was established to enhance national security by providing a reliable and supportable supply of equipment and other necessary items to the various army units.

To achieve its objective, the logistic unit has adopted the following principals;

- **Logistics intelligence;** Commander must have accurate and timely logistics information in order to provide effective logistics support.
- **Objective;** Logistics endeavors must be directed toward a clear and attainable objective.
- **Generative logistics;** The professional application of initiative, knowledge, and ingenuity, and innovative exploration of technical and scientific advances are fundamental to the generation of logistics systems improvements.
- **Interdependence;** Logistics system efficiency requires integration with other functions of the system.
- **Simplicity;** Simplicity is essential at all levels of the logistics system.
- **Timeliness;** Logistics support must be provided in the right quantity at the proper time and place for accomplishment of the mission.
- **Impetus;** The impetus of logistics support is forward to support the combat mission.
- **Cost-effectiveness;** Efficient management of resources is essential.
- **Security;** Security must be maintained to preserve resources and ensure sustained combat capability [Ref. 1: p. 34]

2. The Army Logistics Unit Organization

The Army logistics unit is organized into three levels, each responsible for certain functions. The next two sections will outline the organizational structure and the functions of the logistics units.

a. Organizational Structure

The organizational structure is divided into three levels. The top level is the wholesale echelon which includes depots, maintenance points, plants and factories associated with special army activities controlled by the Army headquarters. The intermediate echelon, i.e. retail sale echelon, is the next level and serves as an interface between the top level and the direct support and use echelon. It is responsible for providing general support function. The third and final level is the direct support and user echelon which includes field units and provides direct support functions to the various units in the field.

b. Support Functions

Logistics Units perform the following functions

- **Supply**; which includes procurement, distribution, maintenance while in storage, and salvage of all commodities necessary to equip, maintain and operate the armed forces.
- **Maintenance**; which focuses on repair and restoration of fielded weapons and equipment systems. Maintenance is classified into three level, i.e. unit, field and depot.
- **Transportation** of troops and supplies.
- **Services**; including food stores, clothes stores, laundry, grave services, fire fighting, etc.
- **Facilities**; including real properties such as depot, maintenance factories, and barracks, etc.

3. Management Issues

Four managerial issues were identified by the Korean Army Logistics Command as critical to the performance and execution of the logistics tasks. These elements are;

- Highly reliable communications between the logistics units.
- Retention of sufficient defense resources.
- High speed movement of combat support supplies.
- Utilization of automatic data processing systems in order to effectively manage the process. [Ref. 2: p. 636]

This thesis is an effort to support the four elements, i.e. the use of data processing in developing a system to improve the retention of sufficient defense resources; to reduce communication complexity and paperwork; to decrease the response time for requests; and to allow management more time for important decisions.

The next sections outline a specific management problem followed by, in later chapters, a proposed implementation of a computerized information system, within the constraints of the Korean data processing environment system.

B. SELECTIVE MANAGEMENT

1. Overview

Material management involves thousands of individual transactions each year. To do their job effectively, material managers must be able to effectively use their time, concentrate on critical items and avoid the distraction of less critical details.

In reality, it is difficult to achieve this without the following prerequisites;

- A classification system to identify critical items.
- Well established and structured procedures to deal with those items that are considered less critical.
- A computer system which assists managers in carrying out the less critical activities and which can identify changes that may effect an activity classification.

The Logistics unit has adopted a classification system which satisfies the first prerequisite above, i.e. inventory control procedures that isolate those items requiring precise control from those items that do not.

2. The ABC Classification system

This system is based on the fact that only a small percentage of inventory items account for most of the total inventory value. Thus considering the cost of management time, it is more cost effective to purchase a sufficient supply of low cost, low demand items and maintain little control over them.

Before discussing the classification system, it must be stated that before an item is given low priority classification, it is evaluated by the Army to determine how critical it is for combat. If it was classified as combat critical, it must be treated like high priority item even if its demand classification is still low one. Thus a low priority item has to satisfy two criteria; an item shortage will not disrupt the combat operations and its annual demand in terms of must be low.

The selective management, more commonly known as the ABC system, was adopted by the Korean Army to meet these supply function considerations in material handling. Each item is given one of three classifications, i.e. A, B or C (see Figure 2.1). Class A consists of items whose dollar value of total annual demand typically accounts for 50% of the total dollar value of the inventory, while representing only 7% of the

number of inventory items. The B class consists of 18% with 35% of inventory items. The C class consists of items whose annual dollar value accounts for only 15% of the total dollar value of the inventory but represents 65% of the inventory items.

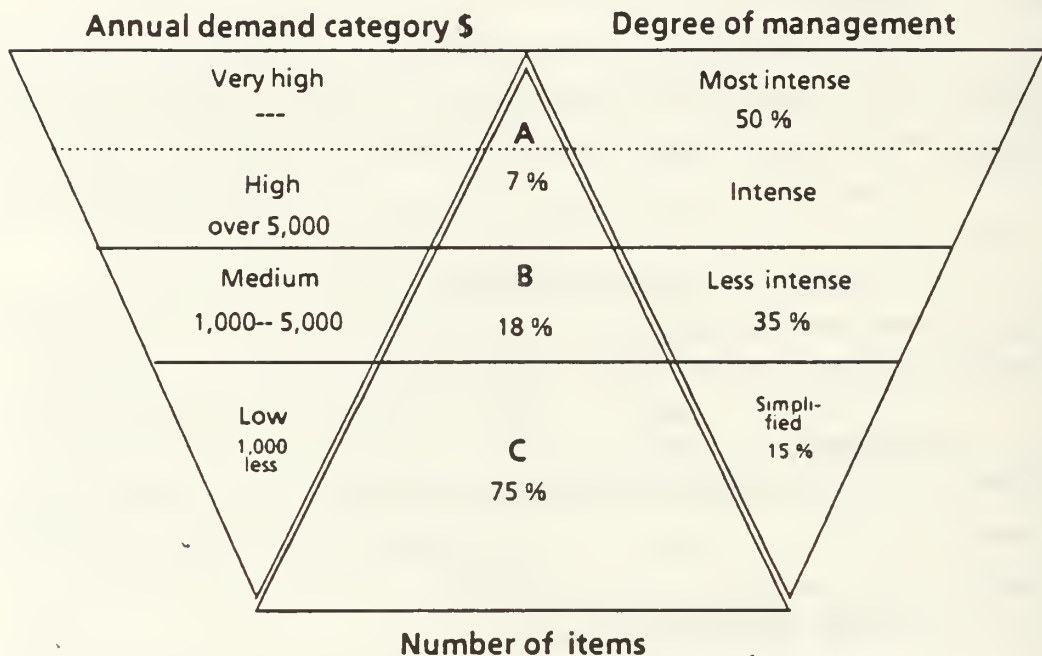


Figure 2.1 ABC Classification.

The entire inventory is listed in descending order from the largest value of the annual demand to the smallest and break points are between class A and B and between class B and C.

The ABC classification exists to direct attention to those inventory items that represent the largest annual expenditures. If inventory levels can be reduced for class A items, a significant reduction in inventory investment will result.

"The purpose of classifying items into groups is to establish appropriate levels of control over each item. ABC analysis is useful for any type of independent demand system (continuous review, periodic review, and so forth). With the periodic system, the ABC analysis can be subdivided so high usage items receive a short review and low usage items receive a much longer review."
[Ref. 3: p.439]

As previously mentioned the degree of control is classification dependent, thus class A items require and are given the greatest attention while the class C items are paid the least attention. Class A items are managed using an economic order quantity

model, A review of the inventory position would occur each time an item is issued to a customer. Class B items could use an economic order quantity (EOQ) based requisitioning objective. Class C items require no special calculations, since they represent a low inventory investment. The order quantity might be a one year supply with a annual review of the inventory position.

The success of the ABC system is dependent on several factors, among them is the validity of the assumption that class A items constitute a small percentage, 7%, of the total items and around 50% of the total cost. Figure 2.1 depicts the distribution of the cost and quantity for the inventory in the Korean army.

The system proposed in this thesis applies continuous review on class A and some of class B items, which are included in the authorized storage list (ASL). ASL will be discussed in section three inventory control means.

C. THE EXISTING SYSTEM

1. The Planning-Programming-Executing-Evaluating System (PPBEES)

The Planning-Programing-Budgeting-Execution-Evaluation System (PPBEES) is a newly developed system for the national resource management in Korea. It evaluates managers based on their budget expenditure and their material readiness condition.

A number of plans were developed to implement this system. For instance Accounting System for Fund, and Fund Management Comparison and Evaluation System.² they had less success than originally envisioned. It is my opinion that the effectiveness of the system can be increased by widening its implementation to include Army divisions and by establishing a vehicle for division managers to easily obtain accurate information needed in managing inventory. At the present time performance and operating statistics must be manually prepared, which is a time consuming process normally avoided by managers. The proposed system allows managers to acquire the needed information in a timely fashion and to support both planning and internal performance audits.

The PPBEES measures performance based on a subset of the supplied items. The following sections outline the division supply system and the items included in the PPBEES system.

²These two names are literally translated from Korean.

2. Resource flow in the infantry division

The resources (or fund) flow in the division is divided into two major categories, materials and cash. The materials are about 55% of total value, and are the subject of the proposed computer system in this thesis, while the remaining 45% is cash.

The materials issued to the division are divided into three groups resource control number (RCN),³ automated supply items, and non monetary valued items, see Table 1

TABLE 1
BUDGET RATE OF MATERIALS

Item	Contents	Percent	
		Total	Line item
PPBEES Materials	Parts, Gasoline, Medical Equipment, General Supply	15	30
Automated Issue Items	Food, Clothing, Heating Fuels, Tactical Construction Materials, Ammunition, Assemblies, End items	40	70
Non Monetary Valued Items	Clothing(Officer,NCO) Field manual, Technical manual Target	Un-known	Un-known

The RCN grouped material constitutes around 15.3% of the resources, i.e. 30% of material, and are issued by the Logistics Support Command with RCN. These are the only items tracked by the PPBEES system and used in evaluating managers' performance.

The remaining materials, 40% of the budget, are issued automatically without user's request. We can anticipate that this results in excess stock in some units while the other unit run out. The division has no evaluation system for these automated issued line items. [Ref. 4: pp.194-196]

³RCN (Resource Control Number) is the numeric symbol which is used for accounting, execution of supply activities, that connects the cash budget and material supply.

The division has only basic manual means for recording material consumption. The inventory report card contains only the present amount of stock for each item in the organization unit. The division should have a system to record material consumption and evaluate support performance.

In the present PPBEES there are no means to evaluate the ranking of the units in a given period, as well as a number of other problems which make an automatized system highly desirable. For example;

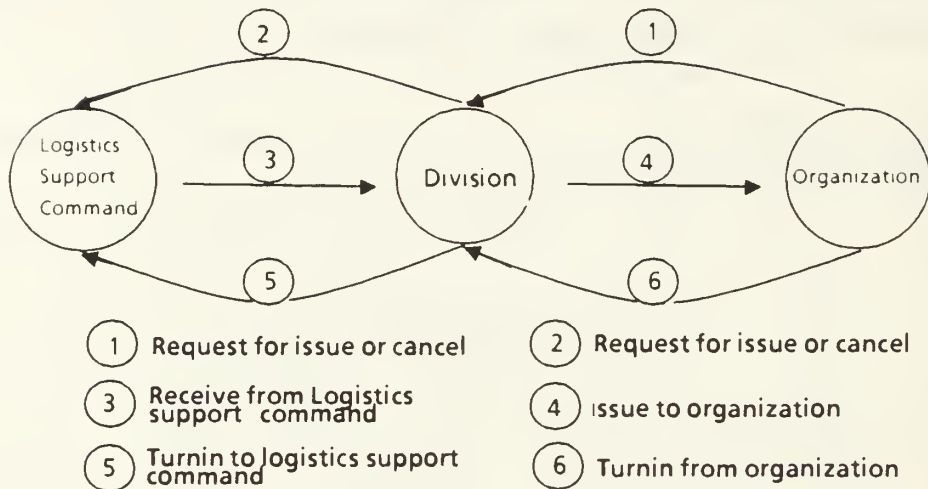
- For certain items, the logistics support command sets a ceiling (maximum allowance) on the number of units of the item which may be consumed by a division each calendar year. The performance of the division with respect to this item is judged based upon the fraction of the maximum allowance which is actually used during the year. Obviously divisions with larger maximum allowances use items more freely, while divisions with smaller maximum allowance may suffer from lack of material. This may induce a division commander to delay the maintenance of equipment, to continue to use substandard material, and to occasionally make inappropriate substitutions.
- There are too many elements to compare. For each RCN item there are 20 criteria used in the evaluation. Each division has 20 units and each regiment has 23 companies. Division managers do not have the required resources to do this comparison manually.
- The system requires too much paperwork. Headquarters must calculate costs by equipment and by subordinate units. Both are difficult manual efforts which may include significant errors.

3. Inventory operation and control in divisions

The supply procedure in divisions is shown in figure 2.2 The principal actions between supplier and customers are; requests for issue; turn-ins; and cancellations. These actions take place between a unit and a division and between a division and the logistics support command. For example when a company sends a "request for issue" to a regiment, the regiment passes it to division. The division issues the item if it is on hand; if not, a request is sent by division to the logistic support command. The logistics support command processes the requisition through their computer system, and generates an issue list containing all the items requested by the division. After the division receives the list and material, all due-outs to division customers are filed and the remaining material is stored pending the next request. All requests for issues from lower units are handled daily, The proposed system follows this basic procedure.

The control measures in these procedures are the authorized storage list (ASL), requisition objective (RO), safety level (SL), and order shipping time (OST).

A. Supply procedure from division



B. Current Transaction record procedure

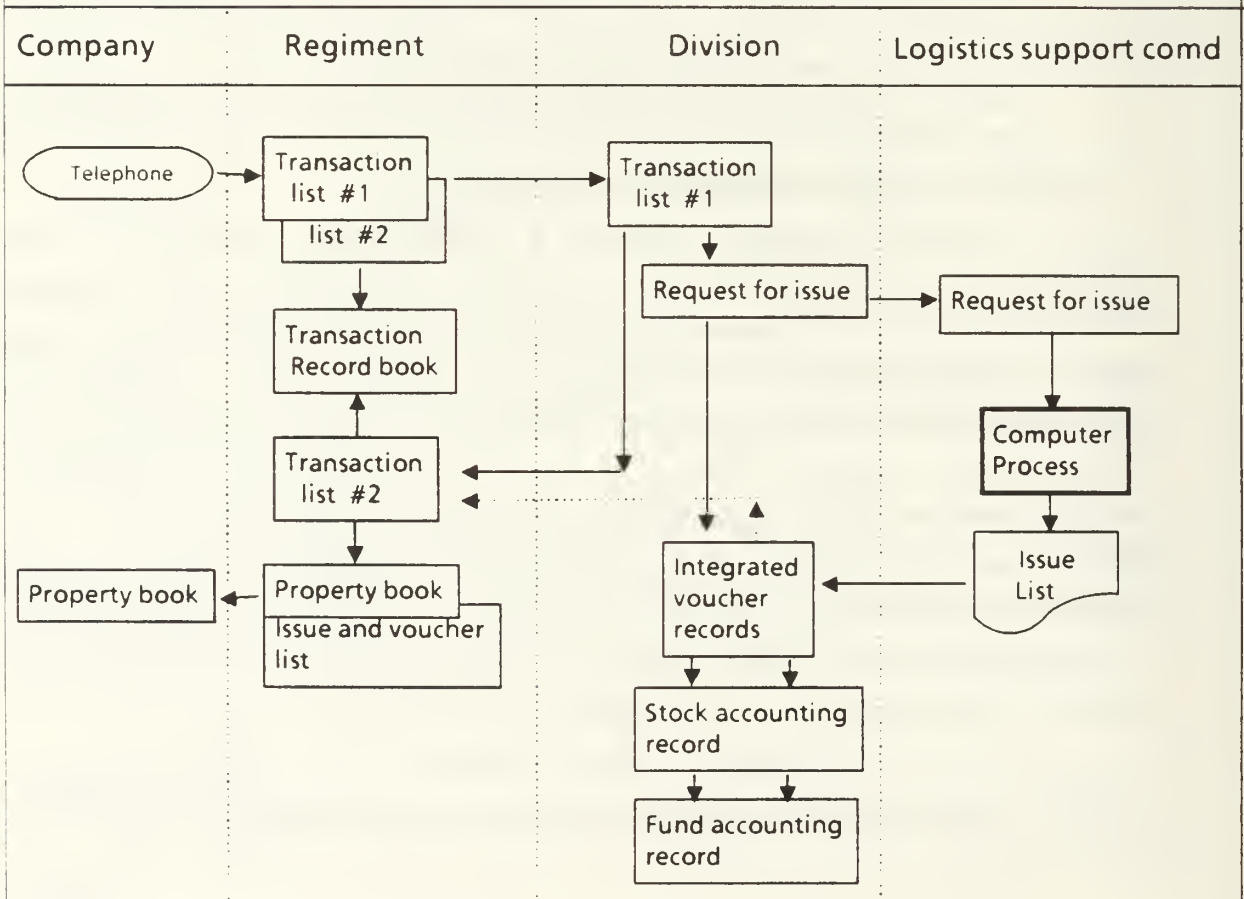


Figure 2.2 Division Supply Procedure

a. Authorized Storage List (ASL)

Authorized storage list defines all supplies which can be stored by the supply unit to meet expected demand. This is based on selective management similar to the ABC classification. The ASL considers economic and support performance objectives. If support performance was the only consideration, all items will be stored. This obviously would be very costly, would increase the difficulty handling the material and decrease the maneuverability of the unit. On the other hand if we consider only economics the unit performance would be degraded and there would be a serious increase in unnecessary stock. Therefore the stock should be the minimum required to perform the mission.

To get the minimum required stock, we have to consider measures of effectiveness (MOEs): Let's assume that total cost = $f(x)$ and support performance = $g(x)$. We have to minimize $f(x)$ and maximize $g(x)$, but as x increases both $f(x)$ and $g(x)$ increase. There is conflict ! So what is usually done is:

1. Choose a specific minimum value for $g(x)$, ie $g(x)$ must be greater than this value (call it "b").
2. Minimize $f(x)$ subject to $g(x) \geq b$.

A ranking similar to the ABC system was developed for the demand frequency of each item. As shown in Table 2 --(A) , 85% of total demand frequency includes only 15% of all items required by the supported unit; even if we increase this latter percentage to 50% and reexamine the associated fraction of total demand frequency, the increase of demand frequency is only 3-4%.

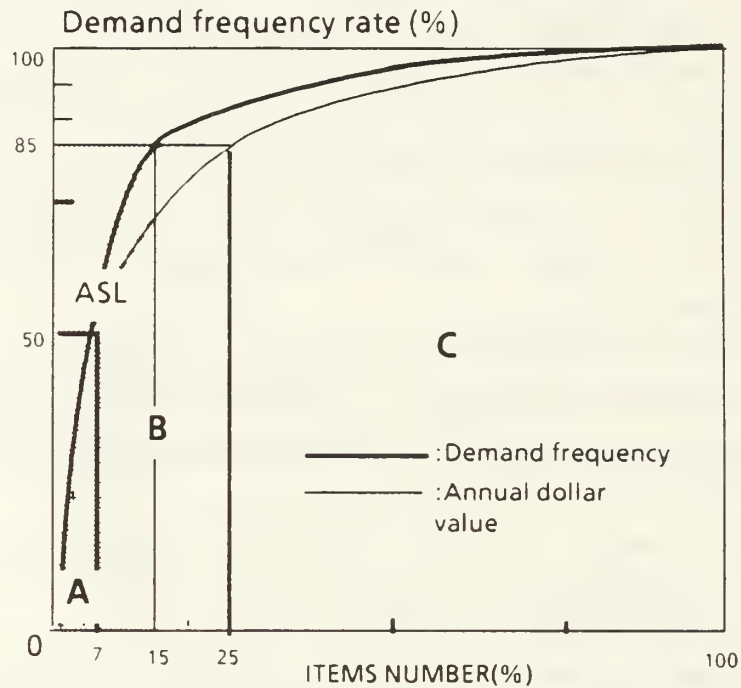
This fact suggests that by storing 15-20% of all items it is possible to meet 85% of the support requirement. This is the basis of ASL, ie satisfying 85% of all requisitions. The achieved DFR is the ratio of demand for ASL items to the total effective demand. Total effective demand is the demand of all items less cancellations.

ASL is divided into 6 groups: active items, stand-by and essential items, Prescribed Load List (PLL) of Organization Unit, the supported units ASL, direct exchange and repair parts for new equipment, and substitution of maintenance.

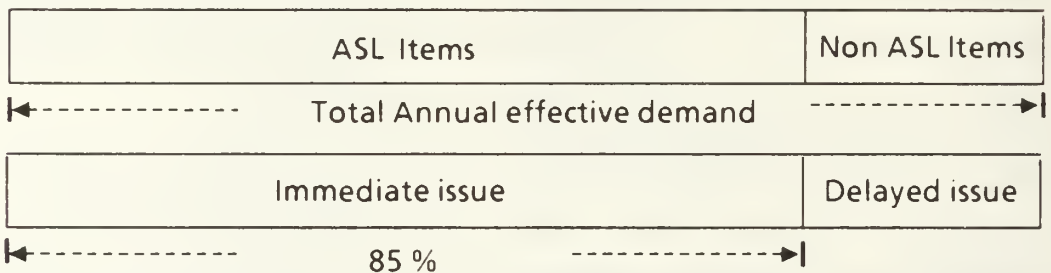
- The active items are the items included in 85% of the demand frequency rate.
- Essential items for any future emergency are decided by the Army Commander.
- Prescribed load list of organization unit are repair parts and tools intended to give worth of support 15 days and must be stored in each organization.
- When two or more items have functional and physical characteristics that cause them to be equivalent in performance, reliability and maintainability, only one of the items will be on the ASL

TABLE 2

A). DEMAND FREQUENCY RATE



B). AUTHORIZED STORAGE LIST



$$\text{Demand flexibility rate} = (\text{ASL} \div \text{Total effective demand}) \times 100$$

C). ANNUAL DEMAND FREQUENCY FOR ASL

Chemical Items	6times	Quatermaster Items	6times
Engineer Items	3times	Communication Items	3times
Ordnance Items	6times	Transportation Items	6times
Medical items	6times		

When stock changes or when catalog data are updated, all additions or deletions are made in the ASL. Other changes include inter-changability and substitution between items. The deputy chief of staff logistics (DCoSLOG) in the Army HQ reviews the ASL semiannually for potential range reductions. Inclusion on the ASL depends on the minimum number of times an item is ordered as shown in Table 2-C.

b. Requisition Objective (RO)

Each ASL item must have an RO recorded in the stock accounting record. The RO is the maximum quantity of the item authorized to be on hand and on order at any time. Retention of assets above the RO is authorized under certain conditions. The RO computations are made in a days of supply (DOSs) mode or economic order quantity (EOQ) mode.

(1) *Days of Supply (DOS)*. The DOS mode will be used by nonautomated accounts when computing stock for items that are critically short, seasonal, highly perishable, or have a shelf life of less than 3 years. A DOS RO will be computed at least semiannually, or when the balance on hand is equal to or less than the reorder point (ROP), or when the balance on hand equals zero.

The DOS RO is the sum of the operating level (OL), safety level (SL), and order shipping time (OST) in days; multiplied by the quantity demanded during the control period (one year for division), divided by the number of days in the control period. The ROP is the sum of the SL and OST in days, multiplied by the quantity demanded during the control period divided by the numbers of days in the control period.

(2) *Economic order quantity (EOQ)*. The EOQ RO is the sum of the EOQ and the ROP quantities which will minimize the total variable cost of stock for a specified performance goal. The performance goal is based on how essential the item is. The amount of stock may be constrained by mobility and fiscal limitations. The total variable cost consists of order cost (replenishment and wash actions), change cost (cost to add and delete an item), cost to maintain an item in stock, storage holding cost, and the implied shortage cost. The implied shortage cost is not a measurable cost (to calculate this cost refer to the Appendix A statistical consideration.)

The implied shortage cost will be used in automated systems as a variable to adjust a stock level to meet a specified performance goal. Stock criteria will be variable when the EOQ is used by automated systems. The EOQ mode will be used by automated accounts (higher than division level so far).

c. Order shipping time (OST)

OST is used computing both EOQ and DOS. It is the average number of days that elapse between the document date of the requisition and the date the receipt is posted to the stock accounting record.

When OST is not available, as in the case of the manual system used in DOS, it has to be computed. The computation is based on the average OST of the six most recent replenishment receipts and rounded to the next higher number of whole days. In computing OST, requisitions are excluded if they have long delays from wholesale backorder, unusual circumstances, or lack of funds. A cumbersome manual process!

OST is updated each time the RO is recomputed. Because of the fact that most of the system is manual, updates to the OST is done annually.

4. Current documentation and report forms

The documents maintained by division logistics departments are classified by contents and include;

- Inventory status for each item
- Evaluation worksheet for the PPBEES
- Documents for TAMMS⁴
- The record of consumable item consumption

These report mainly focus on the historic data and are difficult to use in computing statistics and information needed by managers. For example the current documentation is inadequate to use in material planning. Although it shows the current stock level of items, funds, etc. This data can not be used by a computer and is difficult to retrieve, sort or manually manipulate to calculate the needed information.

The format of the recording form is inconvenient. The documents or forms should be able to prove the transaction history and also be user friendly. In the current documents, transactions are recorded by date sequence, when any cancellation of requisition occurs, it is very difficult to locate and correct all previous records. Records written in pencil are hard to use as vouchers.

⁴TAMMS is a abbreviation of The Army Maintenance Management System which contains the historic records of maintenance, and operation of all equipment. This data will be used in requisitioning of the new equipment requisition.

D. CURRENT USE OF COMPUTERS FOR LOGISTICS IN ROK ARMY

1. Overview

In March, 1967 the first computer was introduced in Korea to assist with a census of the Korean population for the Economic Planning Board. The Korean Army installed its first computer system in the Army HQ for the management of military personnel in 1972. The next year another computer was installed at the Logistics Command to assist with logistics management. Subsequently several computer centers were established. All these computer centers are directly controlled by the staff of the Army HQ. [Ref. 5: p.14]

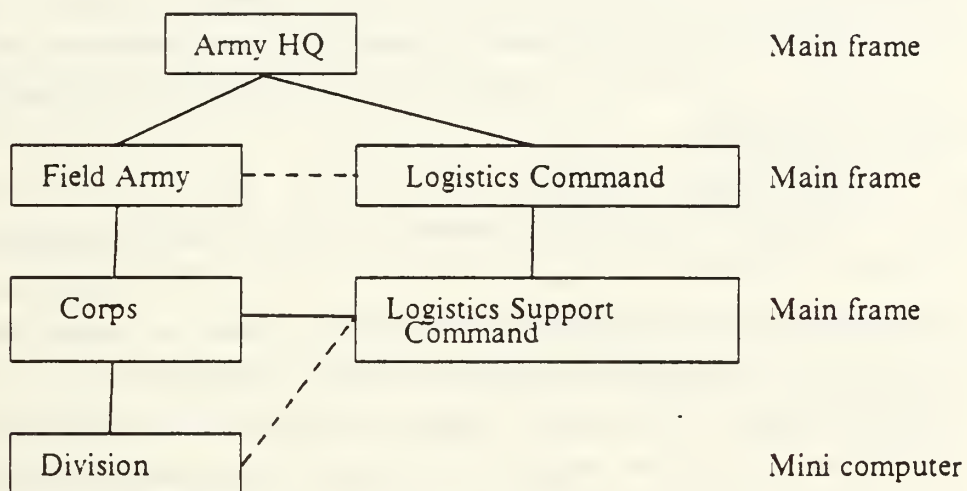


Figure 2.3 Army Logistics Computer System.

In the late of 70's, an integrated software development center was established at the Central Automatic Data Processing Center (ADPC) to develop software for the mainframe computer used by the logistics command and Army HQ. The second center, the Logistics Management Information Center, was established to support MIS development activities in the G-4 of the Army. It endeavors to improve MIS capability and has gained the attention of high level managers.

The hardware installed in the various computer centers is different. IBM 370, and UNIVAC 90/30 and 1100 series machines are very common. These are batch system and are not connected with each other. The application software, which is run periodically by users, was developed using old technology, i.e. flat file system.

The file system and the fact that the computers are not connected results in a high degree of redundancy. Each computer center, even those under the control of the Department of Computers at Army HQ, has several files containing the same data elements. For example the Army personnel system, payroll system, and medical system all contain common data elements which are updated independently, a potential source of data integrity.

A second problem which has affected the development of the data processing field in the army is a shortage of qualified personal.

Recently, high level managers have recognized the need for the standardization of hardware and unification of application software. This has resulted in an ongoing effort to bring the data processing systems in Korea to the leading edge of technology. As a result, a number of mini- and microcomputers have been installed at the division level.

2. ADP Support to logistics

It is necessary for the commanders to have adequate forecasting capability for the effective command and control his unit. The use of ADP systems has significantly increased the commander's visibility and has had an effect on logistics operations. The Automatic Data Processing Center (ADPC) within the logistics structure has provided significant support. The ADPC, dedicated to logistics operations, supports its own internal functions such as stock control within the responsible area and routine jobs such as reports generation for higher commands. Additionally the center provides to other departments the logistics information. In the division level, reports are manually generated.

One concern of this thesis is an important report generating function of ADPC; the inventory status report. This report is presently created by division G-4 for logistics support command manually.

This reporting system will be designed in this thesis to provide up-to-date accurate inventory status data for major items pertaining to each division. In addition to the inventory status report there are many other reports which are needed to control the inventory. These reports provide information to the division commander and higher logistics support units so that readiness can be evaluated. These reports also indicate the shortage and overage of material and, when integrated at higher levels, allow the command to determine new procurement needs, prepare budgets, redistribute assets and take disposal actions.

E. SUMMARY

This chapter presented a description of the existing inventory control system and data processing capabilities. Several issues, stated in this chapter, were paramount in the decision to develop the proposed inventory control system. For example;

- The existing system has limited capability in the development of effective material management.
- The PPBEES implementation will benefit from a computerized Management Information System.
- The documentation in the present system is cumbersome and inadequate for managers.

A database management system can be used to resolve these problems. A well developed one will provide the information needed, maintain records of transactions and exchange the information with other computers.

This system has to be user friendly and reduce to an acceptable minimum the dependency on the data processing personnel. Its implementation, including installation, must be simple and must use the microcomputers available in the various units.

The problem the system resolves is well structured to insure the system's success. At the same time the system design has a degree of flexibility to expand at a later point. The existing system is labour intensive and the proposed system will save managers precious time for more important activities.

The system described in this thesis was designed with these facts in mind. As such it resolves a problem by using microcomputers, and a user friendly interface, and allows for electronic communication of data.

III. RELATIONAL DATABASE MODELS

A. INTRODUCTION

The proposed system is a relational database management system which uses dBASEIIIplus. In this chapter, a brief description of relational database models is given.

B. WHAT IS DATABASE?

A Database is basically a computerized record keeping system, that is, a system whose overall purpose is to record and maintain information for future retrieval. D.R. Howe, the author of *Data Analysis for Database Design*, defined a database as "*a collection of non-redundant data shareable between different application system*". He extended his definition by saying that '*non-redundant means unnecessarily duplicated data adds no new information*', sharing data as '*a multiple usages in multiple applications*'. [Ref. 6: P.1]

A database should be structured so as to provide a foundation for future application development. It can be manipulated into information for management purpose.

A database model is an abstract representation of data. It defines the way that data items are organized and related. There are two major classes of database representation, the physical and logical models. The physical model represents the actual structure of the data in the computer. The logical model represents how a user perceives the data organization. For example, in an inventory system, we may have the following case;

A number of parts are located in a number of warehouses. shown on Figure &str-A.

The physical structure may consists of the following files

- A part file which consists of the part number and all the attributes which specifically describe this part, such as part name, price, weight, and the vendor supplying this part (assuming that a part is supplied by one vendor only), etc.
- A warehouse file which consists of a list of all warehouses and those attributes unique to each warehouse.
- A relationship file which consists of the attributes that describe the relationship between parts and warehouses, e.g. number of units of part 1234 (QUANTITY) in warehouse A-12 and location of the part in this warehouse (bin 27AC)

B. Manager's View

W_NO	P_NO	VENDR
A-12	1234	GM

C. Worker's View

P_NO	W_NO	QUANTITY
1234	B-11	120

A. Relationship

P_NO	W_NO	QUANTITY	BIN-NO
1234	A-12	100	27AC
1234	B-11	80	27BC

P_NO	PRICE	WGHT	VENDOR
1234	12.00	12	GM
1111	15.00	30	HD

Parts file

W_NO	LOC	DIRECTOR
A-12	V1C10	Reagon
B-11	V2C9	Bush

Warehouse file

Figure 3.1 View of database.

The logical structure is a user view, and as such it is dependent on the user. For a warehouse manager, the logical view may be similar to the one shown in Figure 3.1-B, ie. the warehouse data and the parts available in that warehouse and the vendor of each part.

A second user view may be that of a worker on a machine. In this case the person is interested in seeing the part data and attributes as well as the warehouses where this part is. He is not interested in knowing the vendor or the location of the part in a warehouse.

This type of file structure facilitates adding new views based on user requirements without changing the physical structure of the files and reduce the number of redundant elements and the update effort. Programs share the data instead of having the same data duplicated for each program, creating an update nightmare.

The database management system is responsible for managing the physical storage of data. Thus if a data element physical characteristic changes, no programming changes are required since each program requests a data element by name and is, to a great extent, independent of the element's physical characteristics.

C. RELATIONAL DATABASE MODEL

The database model discussed in the previous example is called the Relational Model. A basic quality of the relational model is its simplicity.

It was introduced by Dr. E. F. Codd in a seminar paper in 1970. The paper have been written concerning the most appropriate way to express relations. He stressed the independence of the relational representation from physical computer implementation such as ordering on physical devices, indexing, and using physical access paths. Although the relational model has many desirable characteristics, it was, until recently, a subject of theoretical interest only. In the last few years commercially viable database management system (DBMS) became available and in the early 1980s, several important DBMS products were introduced. SQL DS (vended by IBM) and ORACLE (vended by Relational software Incorporated) are two examples. Since these announcement, the relational model has come to be of greater practical significance.
[Ref. 7: p. 242]

The following definitions introduce a number of terms that will be used in describing the proposed material control model.

1. Relation

A relation is a table of data (a file) and consists of rows (tuples) and columns (attributes). The data table provides a simple data structure.

		Attribute	Attribute				
		↓	↓				
		SN	NM	UNIT	ONHAND	UNITCOST	CLASS
Tuple →		1111-11-111-1111	Test item1	Ea	120	12.90	1
Tuple →		1222-22-222-2222	Test item2	Ea	90	15.00	1
		1333-33-333-3333	Test item3	Ea	100	25.00	1
		4444-44-444-4444	Test item4	Ea	190		4
			~				
		9999-99-999-9999	Test item4	Ea	9999	100	9
Records							

Figure 3.2 Relation of property.

Figure 3.2 shows an example of a table (or relation); the table contains six columns (each represents an attribute) and five rows (tuple, each represents a specific record). The intersection of each row and column in the table contains a value. For instance,

'Test item1' is an attribute value of the attribute type "NM" for the record belonging to stock number 1111-11-111-1111.

There are a number of constraints that must be observed when building tables;

- The ordering of rows is not important because the rows can be interchanged without affecting the information content of the table.
- The ordering of columns is not important either, for the same reason.
- Each row and column intersection contains a single attribute value. Multiple values are not allowed.
- Each row in a table must be distinct; no two rows can have the same attribute values throughout. (The significance of this rule is that a row can always be uniquely identified by quoting an appropriate combination of attribute values.)

An attribute value may be null, but a null value does not mean that the attribute is blank, e.g. the unit cost of test item4 may be entered in the table after the record has been created.

Each attribute has a domain, a set of values that the attribute can have. For example, the domain of unit cost is a positive eight-digit number with 2 decimal points.

2. Keys

We want to be able to identify each tuple (record) in a relation by the value of at least one of its attributes. In Figure 3.2, the stock number (SN) is a unique identifier since no other row may have the same stock number.

If there is no one unique attribute, then a combination of more than one may have to be used. The table data structure shown in Appendix C includes examples of keys which have more than one attribute.

3. Record relationships

The essence of a database is the representation of record relationships. The relationships can be specified in a variety of ways.

The relationships are identified intuitively. The designer considers potential relationship among records that have been defined. A relationships may exist among three or four or more records. For example with records of the proposed system, MASTER records have many property records, while one PROPERTY record has only one ASL record. Many STOCK-OUT records have many MASTER records.

4. Database Management System

A database management system (DBMS) is a software system which performs the functions of defining, creating, revising, and controlling the database. It provides facilities for retrieving data, generating reports, revising data definitions, updating data, and building applications.

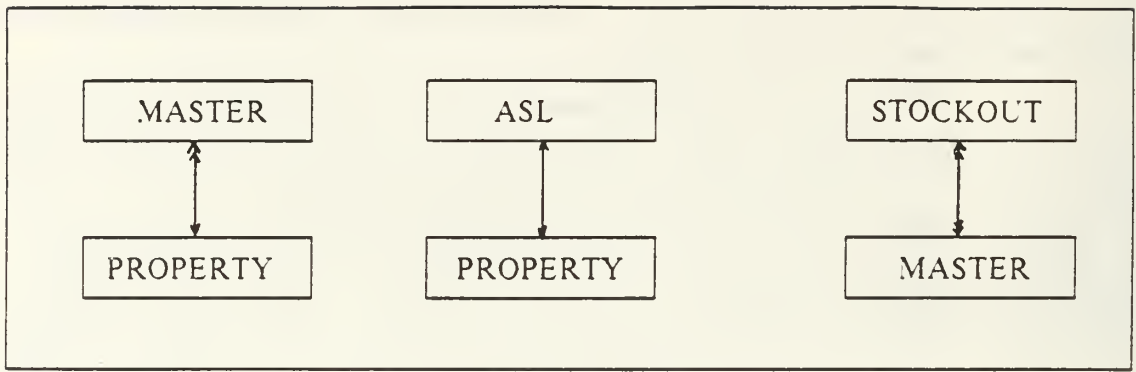


Figure 3.3 Record relationship.

Several relational DBMS are available. The one used in developing this inventory control system is Dbase III Plus. The version used is the Microcomputer version. However, an extension which includes SQL, a facility which allows sharing of data with mainframe computers is under development and will be introduced in the very near future.

DbaseIIIPlus is the market leader among DBMSs in the United States. Several supporting products have been introduced such as compilers to increase execution speed and application generators to optimize the code execution. It has network capabilities which allow sharing of data, and security procedures not available in the single user version. It also has its own programming language, which facilitates the development of user friendly interfaces.

IV. THE PROPOSED SYSTEM

A. OVERVIEW

This chapter describes the existing manual procedures and the proposed computerized system.

B. MANUAL SYSTEM

Presently, a logistics unit in a division receives a request for issue of an item, an item turn-in, or cancellation of a requested item from lower level organizations. The same set of requests occur between the division and the LSC. Transactions are recorded in property books and this generates a second type of process, that is the editing of erroneous entries.

Transactions are aggregated to produce management reports, which are used during the analysis for the next planning cycle and for evaluating the performance of managers.

Two sets of books exist. the first book is the transaction log in which records of each request are kept. Each entry includes the transaction number, date, item ordered and quantity. The second set is the property book which has a page for each stock number and the on-hand quantity of this item. When a request is satisfied, both sets of books are updated.

C. THE PROPOSED SYSTEM STRUCTURE

As requests arrive at the division, either from outside entities or internal personnel, the end user enters the request data using a screen identical in its format to the request form. Entries are edited by the system for errors using hard coded criteria, and stored.

The system consists of several files and modules. The Batch and Master files are similar in structure to the Transactions Record Book, while the Property File and the ASL file are similar to the Property Book. An item is included in one of these latter files based on its classification and whether the item is PPBEES trackable or not. The Stock Out file combines elements from both books and the Customer file contains customer's attributes such as address, customer ID, zipcode, etc.

Modules in the system perform several functions. For example, when transactions are processed the system updates the quantity on hand, creates a transaction record, the completion date of a transaction is generated and the transaction record is moved to the MASTER file. The date field is used for performance measurement purposes. For example the date field will help answer questions about turn around time for a specific item.

If the stock on hand after satisfying the current demand is less than the reorder point, an attribute for each item, the stock onhand field, is updated to reflect the current onhand amount and a request for issue to the supplier will be generated automatically. When the amount is less than zero, a stock-out record is added to the STOCK-OUT file, and the system sends a delivery delay warning to the user. This process is illustrated in a flow chart in Appendix C.

One issue list per customer is generated, regardless of the number of items requested in one day, under one receipt voucher number. The same is true for a request for issue to LSC. Figure 4.1 shows examples of the issue list and request for issue. The processing of the other type of transactions follows almost the same procedure.

When the supplier issues stock to the division, the stock is examined to determine whether it is new, i.e. initial supply, and if it is, the program will ask the user to enter it in the property file, and in the ASL file if necessary.

The fact that the system does not require more than one entry per request and that it interrogates all of the appropriate files, without end user intervention, drastically reduces the amount of end user time per request.

The system is flexible and can accommodate a number of potential user requests. For example, If a user wants to extend the output information to the type of operation, he can add one attribute to Master file for the operation type, the query program can easily support such requests.

Figure 4.2 depicts the data flow in the proposed system and Figure 4.3 shows the transaction data flow diagram.

D. THE PROPOSED SYSTEM FUNCTIONS

1. Transaction tracking

The main purpose of the transaction tracking is to record each customer's request for inventory planning and management purposes. The proposed system generates reports on the status of each customer's or stock item. Transaction records are saved in the Master File for analysis of trends. The ASL file provides criteria such as Reorder point, requisition objective, and safety level.

ISSUE LIST

Page 1

Date : 11/13/87

From : The 150 Infantry Division
To : The 150 div 1501 regiment

Vocher No:1500QM-0029-7317

No	Stock number	Description	Onhand Unit	Quant'y	Price	Total
1	1111-11-111-1111	Test item1	SORRY! Delay delivery			
2	5555-55-555-5555	Test item5	7976 Roll	300	3	900
3	8888-88-888-8888	Test item8	SORRY! Delay delivery			
					Grand Total :\$	900

Matrrial Management NCO : _____ Date : ____/____/____

Material Management Officer : _____ Date : ____/____/____

REQUEST FOR ISSUE

Page 1

Date : 11/13/87

To : The 3333 Logistics Support Command
From : The 150 Infantry Division

Request No : 1500QM-0014-7317

No	Stock number	Description	Unit	Quantity	Price	Total
1	1111-11-111-1111	Test item1	Ea	105	10	1050
2	8888-88-888-8888	Test item8	Ea	20	22	440
					Grand Total :\$	1490

Matrrial Management NCO : _____ Date : ____/____/____

Material Management Officer : _____ Date : ____/____/____

Figure 4.1 Examples of transaction output.

For management purposes, a manager may ask specific questions on a transaction record, a customer's transaction history, or the status of a certain stock item. However, The system stock item queries are limited to the status of the item. An extension to determine the location of an item may be added with minor changes to a database file. This programming effort may be done by a knowledgeable user.

2. Report Generation

One of the most important functions of the proposed system is to generate reports required by the logistics support command and the internal division manager. The system has a number of hard coded reports and a simple ad hoc reporting facility. The hard coded reports are described in Appendix C.

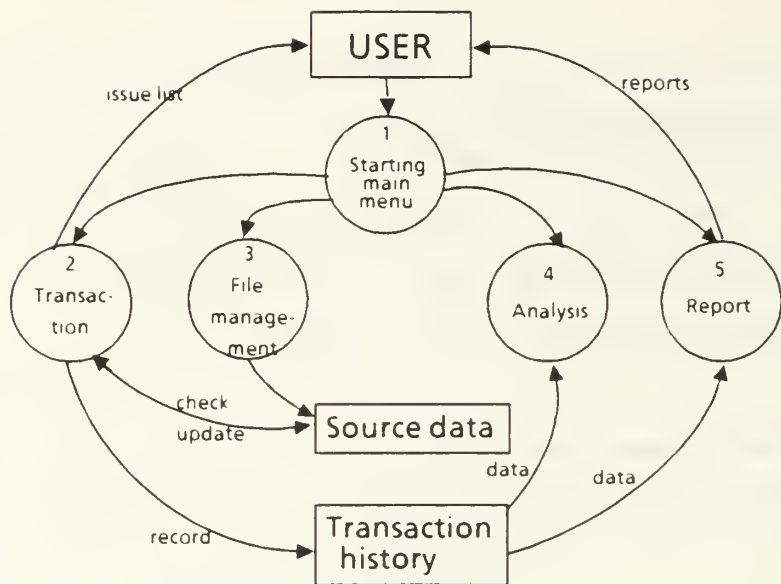


Figure 4.2 Main data flow diagram

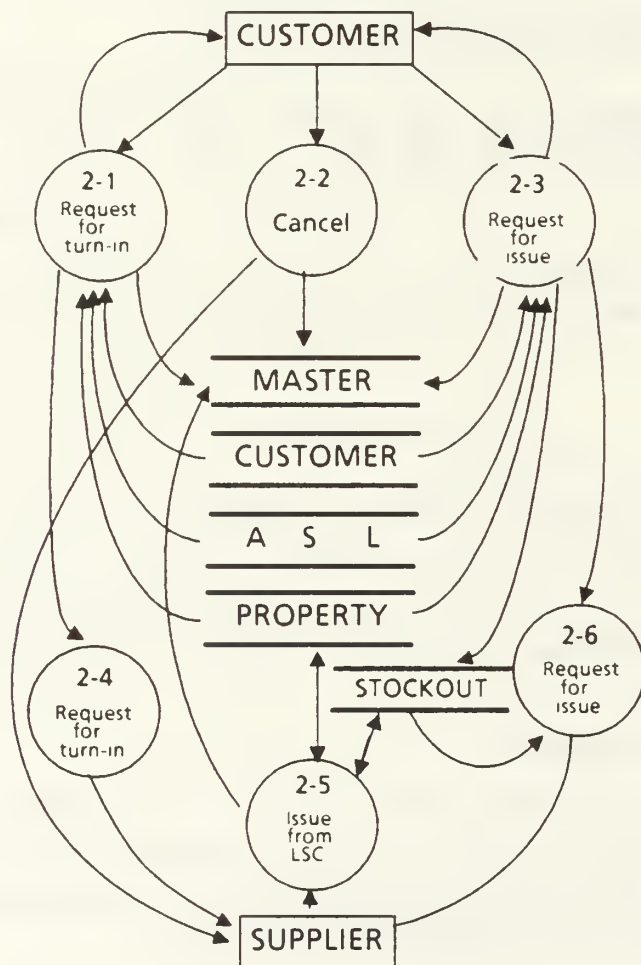


Figure 4.3 Transaction data flow diagram

Reports may be classified, into two groups, internal use and external reports. Internal use reports are those used for performance analysis. The external reports are the transaction output, e.g. the issue list resulted from request for issue of the organizational unit, request for issue, request for turn-in are for Logistics Support Command.

Figure 4.4 shows the data flow for report generation. More reports may be added to the system, if requested, this will bring the system closer to a complete inventory planning management information system.

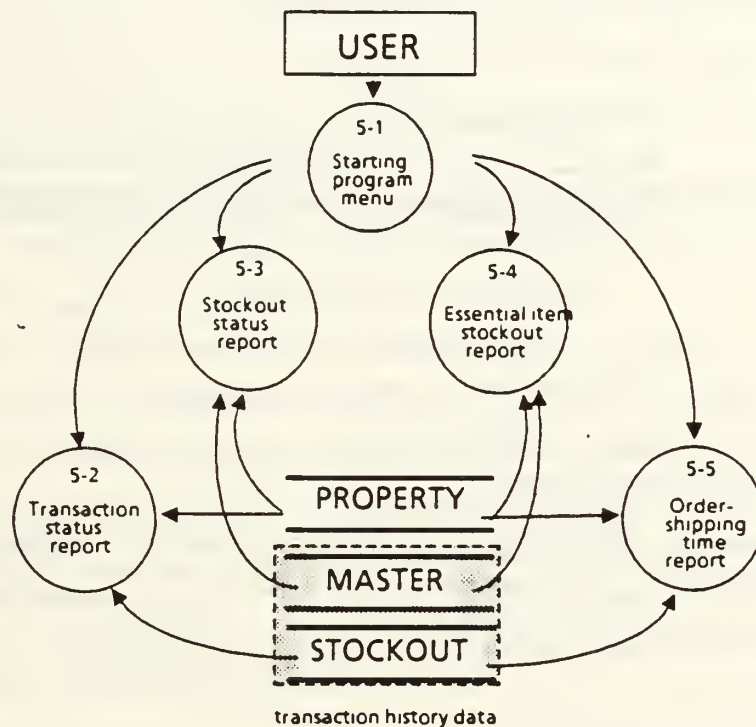


Figure 4.4 Report data flow diagram.

3. Limited analysis

The proposed system uses the fixed order size probabilistic model for determination of reorder point, safety level, and economic order quantity.

The assumptions involved in the proposed system are;

- The demand is continuous and normally distributed,
- Annual demand can be obtained through the transaction history record,
- There are no seasonal effects on demand,
- The lead time is constant,

- Service level for each item is known or determined by managers,
- The stock-out cost is known.

Other statistical consideration are presented in appendix A.

The annual demand is determined by summing the effective demand (total demand – canceled demand) and the monthly demand is calculated by dividing annual demand by 12. The standard deviation of demand is calculated from the transaction record history. The lead time is obtained by computing the arithmetic average of the order shipping times from historical data for a specified period of time.

After all parameters are determined, the reorder point, safety level, and requisition objective are calculated and updated.

I N F O R M A T I O N f o r A N A L Y S I S							
<p>You selected stock number : 8883-88-888-8888 (Test item8, Unit:Ea, Class:8)</p> <p>Purchasing price(P) : 22 \$/Ea</p> <p>Annual Demand(R) : 100 Ea/year</p> <p>Lead time(OST) in Month : 0 Month(s)</p> <p>Ordering cost(C) : 0.00 \$/order</p> <p>Holding cost unit per year : 0.00 %</p> <p>(Select one of these)</p> <p>Stockout cost(If Known) : 0.00 \$/unit Select? N</p> <p>Service Level in year : 0.0000 % Select? Y</p>							
<table style="width: 100%; border: none;"> <tr> <td style="width: 50%; border: none;">[PRESENT] Reorder point :15</td> <td style="width: 50%; border: none;">Safety Level :10</td> </tr> <tr> <td style="border: none;">Requisition objective :50</td> <td style="border: none;">Lead Time : 14</td> </tr> </table>				[PRESENT] Reorder point :15	Safety Level :10	Requisition objective :50	Lead Time : 14
[PRESENT] Reorder point :15	Safety Level :10						
Requisition objective :50	Lead Time : 14						
Is this record right?(Y/N): X							

Figure 4.5 The output from analysis process.

4. Use friendly interface

One of the purposes of this thesis is to design a user friendly computer program. Novice computer users can use it without special training. The proposed system provides a user friendly interface by minimizing the possibility of errors. The user does not have to know about the disk operating system. And a manager who is not familiar with a division inventory system can use it, since most of the transaction processes proceed automatically.

E. SYSTEM LIMITATIONS

The proposed system is not a perfect system. An objective of this system is to improve logistics management capability. This is a difficult task and requires many additional software applications. For example, better system would include accounting and inventory tracking functions. The proposed system addresses just on main logistics function. Constructive comments and remarks are welcomed and solicited.

1. Demand forecasting

Demand forecasting forms the basis of managerial decisions and if it is accurate, managers don't have to worry about safety level or lead times.

The proposed system does not forecast demand. The annual demand quantities are based on one year historical records or as specified period. Managers can retrieve historical demand data in various forms, by item, period, unit, or combinations, and use it in forecasting.

An extension to the software is needed to provide demand frequency information by unit by type e.g. training, exercise, routine job, or special purpose.

2. Statistical analysis of transactions

Statistical analysis was not fully covered in this system. Only two models were presented for analysis; the known stock-out cost model and service level model under normally distributed demand, were presented. There are a number of other models for inventory management. The system should be enhanced by adding various models and allowing organizations to choose the model appropriate for their needs.

3. ASL request

As was mentioned in chapter II, the division commander can ask for additional items for the ASL for his division. The basis of for this request is also described Table 2 in chapter II. The query menu in the software written for this thesis allows the division logistics manager to quickly determine if a particular item will qualify for stockage in the ASL based upon its demand history. The software for this thesis does not, however, include the facility to automatically generate a request to add a particular item to the division ASL.

4. Fund Accounting

The accounting of the fund (of RCN) is important to managers, so the proposed system is partially prepared to do this function. The resource control number (RCN: an attribute of an item) is based on the current PPBEES. While the proposed system, includes all the needed elements, it must be extended to perform accounting functions.

F. ADVANTAGES OF THE SYSTEM

By applying the proposed system

1. The Army can improve one of its logistics objectives: automatic data processing.
2. The system contributes to the PPBEES implementation by generating timely and accurate reports for various supply units.
3. It allows for more effective utilization of personnel time by shortening the administrative process.
4. More accurate records can be maintained.
5. The system can be extended, with minimal effort, to include fund accounting

V. CONCLUSIONS

A. CONCLUSIONS

The proposed system is a user friendly, menu-driven, modularly designed database management system focusing on the supply distribution function of the ROK Army division. It will reduce investment in personnel, and material and generate timely and accurate reports serving division managers. The system may be operated by novice as well as experienced personnel.

To develop this system, the ROK Army logistics structure, current logistical management methods, and problems were described in chapter II. The same chapter describes current computer usages in the ROK Army.

The third chapter includes the structure of the database, its major capabilities, and benefits. The chapter presents the fundamentals of databases and the database management system used in developing the system, i.e. dBASEIII PLUS.

The fourth chapter describes the proposed system: what it can or can't do for the logistics control management in the ROK Army division. The chapter also addresses the benefits of the implementation of the proposed system.

Through the development of this system, the author achieved the primary goal of this study by combining the knowledge of the ROK Army logistics management, relational database, design of the database management system, and programming techniques. The implementation of the DBMS in ROK Army division should contribute significantly to the supply effort.

By using this program, the ROK Army division can reduce, the current volume of paperwork and overtime devoted to routine tasks. The proposed system can save personnel and material resources, thereby freeing these resources to other vital objectives such as enhancement of the combat power.

B. FURTHER STUDY

In this thesis, the author deals only with the distribution function from among many other logistics functions. A more comprehensive system would include accounting, maintenance tracking, and provisions for analyzing transaction records system.

Different inventory control models would be applied to enhance the database management system. Only two of the many models available have been implemented in this thesis.

Networking and interconnecting this system with available mainframe systems is another issue not addressed in this thesis. A final objective is to tailor this system for use in combat situation.

APPENDIX A

STATISTICAL CONSIDERATION

The material in this appendix is extracted directly from Tersine [Ref. 3: 131-165.]pp It is given here because the decision of safety level, reorder point in the proposed system is based on.

According to Tersine, when the demand is probabilistic (not deterministic), it is necessary to minimize the expected total cost for inventory management. If the demand distribution is continuous, the minimum expected total cost expression is obtained by taking the derivative of the total expected cost with respect to the decision variable and then setting it equal to zero.

If leadtime demand is distributed according to a continous probability distribution $f(M)$ then mean leadtime demand is given by:

$$\mathbf{M} = \int_0^{\infty} Mf(M)dM \quad (\text{eqn A.1})$$

and the standard deviation of leadtime demand is the square root of the variance whicch is given by:

$$\sigma^2 = \int_0^{\infty} (M - \mathbf{M})^2 f(M)dM \quad (\text{eqn A.2})$$

Where:

M = random variable for lead time demand

$f(m)$ = probability density function of lead time demand

σ = standard deviation of lead time demand

\mathbf{M} = mean lead time demand

The probability of a stockout for a given item is simply the probability that the demand during the lead time will exceed the reorder point. The stockout probability is the first definite integral of the probability density function of demand during the lead time from the reorder point to infinity.

That is:

$$P(M > B) = P(s) = \int_B^{\infty} f(M)dM \quad (\text{eqn A.3})$$

Where : B = Reorder point

M = Random variable for leadtime demand

The expected stockout quantity during the lead time is the second definite integral of the probability density function of demand during the lead time from the reorder point to infinity.

That is

$$E(M > B) = \int_B^{\infty} (M-B)f(M)dM \quad (\text{eqn A.4})$$

When demand is normally distributed, the reorder point can be obtained from following formula;

$$B = \underline{M} + S = \underline{M} + Z\sigma \quad (\text{eqn A.5})$$

Where

\underline{M} = Mean lead time demand in units

S = safety stock in units

Z = Standard normal deviate corresponding to the desired stockout probability

σ = Standard deviation of lead time demand.

Known stockout cost per stockout event

The proposed system assumes that demand is normally distributed and lead time is constant (that is realistic for many items). It is also assumed that a fixed shortage cost is incurred one time only in each reorder cycle which experienced a stockout situations. This cost is assumed to be unaffected by the number of units or requisitions which are backordered during a reorder cycle. Since the historical distribution of demand is available, the safety stock can be determined by selecting a safety level that results in the lowest expected cost. It is easy to determine the safety stock using this method. The objective is to minimize the sum of the cost of holding the safety stock and the cost of the stockouts. The danger of stockout occurs only during the lead time. There are R, Q lead times of opportunity for a stockout to occur is obtained by taking the derivative of the expected annual cost of safety stock with respect to the reorder point and setting it equal to zero.

Annual cost of safety stock = (holding cost) + (stockout cost)

$$\begin{aligned}
 TC_s &= SH + GR/Q \left(\int_B^{\infty} f(M) dM \right) \quad (\text{eqn A.6}) \\
 &= H(B - \underline{M}) + GR P(M > B) / Q
 \end{aligned}$$

TC_s = expected annual cost of safety stock

$B = S + \underline{M}$ = reorder point in units

S = safety stock in units

H = holding or carrying cost per unit per year

G = backordering cost per outage

R = average annual demand in units

Q = lot size or order quantity in units

$f(M)$ = probability density function of lead time demand

\underline{M} = average lead time demand units

If the derivative of the expected annual cost of safety stock with respect to the reorder point is taken and set equal to zero, the following relationship is obtained;

$$f(B) = HQ GR \quad (\text{eqn A.7})$$

However for normal distribution the optimum reorder point is not obtained from above equation, since the the ordinate $f(B)$ undergoes a change of scale when it is transformed to the standard normal distribution. Thus, if we are to find $f(B)$, we must find here the standard normal distribution has an ordinate of $f(B) \times \sigma$. The standard normal deviate Z for the optimum stockout probability can be obtained directly from the standard normal table.

The optimizing safety stock can be obtained from normal database and the following formulas;

$$Z = (B - \underline{M}) / \sigma = (B - (\underline{D} \times L)) / (\sigma_D \sqrt{L}) = S / (\sigma \times \sqrt{L}) \quad (\text{eqn A.8})$$

$$S = Z\sigma = Z \sigma_D \sqrt{L} \quad (\text{eqn A.9})$$

where

Z = standard normal deviate

$B = \underline{D} \times L + S$ = reorder point in units

$\underline{M} = \underline{D} \times L$ = average lead time demand

σ = standard deviation of lead time demand

L = lead time

S = safety stock in units

σ_D = standard deviation of demand for a time period other than the lead time.

Service level is given instead of stockout cost per outage

Most of the time in the ROK Army stockout costs are very difficult to obtain because of its accounting system. Under these circumstance, using set service level is a more rational way of determining safety stock. A service level indicates a level of ability to meet customer demand from stock.

The establishment of a service level is a subjective management judgement that is based on convenience rather than scientific justification. The choice by management of service level implies a cost attributed intuitively or indirectly to stockout.

Example: If the annual demand for an item is normally distributed with a mean of 8000 units and standard deviation of 1000 units, what should the safety stock and reorder point be if the lead time is 1.2 month? (assume management has decided it is willing to be out of stock in 5% of the order cycles)

From the normal distribution table, $Z = 1.64$ corresponding to a 5% one tail risk of stocking out.

$$\underline{D} = R \cdot L = 667$$

$$\sigma_D = \sigma \sqrt{L} = 288$$

$$B = M_a = (\underline{D} \times L) + 1.64 \sigma_D \sqrt{L} = 669$$

$$S = M_a - \underline{D} \times L = 335$$

Where : R = Average annual demand

$M_a = B$ = lead time demand at acceptable service level in units,

The safety stock is 335 units, and the reorder point is 669 units.

APPENDIX B

DATA STRUCTURE

1. OVERVIEW

There are seven interrelated files in the proposed system. The PROPERTY and ASL file, PROPERTY.DBF and ASL.DBF, are used in the inventory control management. They contain for each item its stock number, its attributes and its inventory control criteria.

The main files for transactions and report and statistics generation are the BATCH MASTER, and STOCKOUT files - BATCH.DBF, MASTER.DBF, STOCKOUT.DBF. The batch file is a depository of all the transactions arriving in one day. At the end of the day, transactions are processed and the demand is satisfied based on the transaction priority, priorities are determined by the unit's Standard Operating Procedure(SOP.) Transactions are moved to either the master or the stockout files.

These three files contain transaction specific data. As detail files, they define only those elements defining a transaction and do not include global or unchanging elements, such as the customer's description or stock item's nomenclature. Global elements are referenced by a unique identifier.

2. DATA FILES

a. Master

The elements in the master file are SN, CI, VN, REQNO, QTY, TYPE, UNITCOST, MISC, DATE, and POSTED. The SN field identifies the stock number of each item on the transaction and is the key to the items file. CI contains the customer number and is the key that link a transaction to the customer file which contains data on customer and supplier. SN and CI link the records to the other source files. VN contains the voucher number of the transaction, and REQNO is the requisition number.

The remaining fields --QTY, UNITCOST, DATE, contain the quantity, unit price, and date of the transaction. The TYPE field contains the transaction type which is shown in chapter II, Fig 2.2 transaction procedure. Table 4 in this appendix includes a detail description of the contents of the various elements.

TABLE 3
DATA STRUCTURE

Database	Field name
Master	(SN, CI, VN, REQNO, QTY, TYPE, UNITCOST, MISC, DATA, POSTED) Primary = SN keys = CI, TYPE Record on all transactions.
Property	(SN, SERIALNO, NM, UNIT, ONHAND, UNITCOST, TVALUE, ESSENCE, CLASS, DATE) Primary = SN Data on characteristics of stock item.
ASL	(SN, ROP, SL, ROOST, RCN, DATE) Primary = SN Data on transaction control measures.
Customer	(CI, CDESC, ADDRESS, ZIPCODE, FUND, EXPEND, PRIORITY, DATE) Primary = CI Data on customer's record.
Stockout	(SN, CI, VN, REQNO, TYPE, QTY, UNITCOST, DATE, MISC, POSTED) Primary = SN keys = CI, TYPE Records on all stockout transaction.
Batch	(SN, CI, VN, REQNO, TYPE, QTY, UNITCOST, DATE, MISC, POSTED) Primary = SN keys = CI, VN, REQNO Intermediate records on transactions.
Normal	(TZ, POFB, POFM, EOFZ) nokey Table of normal distribution.

b. Property file

The property file contains all the information pertinent to each stock number. The first field, SN, is the key for accessing this file. The ONHAND and UNITCOST are quantity on hand and the unit cost respectively. For other fields refer to the Table 3

c. ASL file

The ASL file contains all the information on the control measures for each item. The first field, SN, is the key for accessing this file. RO field is the requisition objective defined in chapter two. ROP is the reorder point, whenever the ONHAND on property file reaches to this point, the system issues a request for issue to the logistics support command automatically. SL is safety level used in calculating the appropriate requisition objective.

d. Customer file

The customer file, CUSTOMER.DBF, whose structure is shown in Table 3, contains all the information on both customers and suppliers. The customer ID (CI) is the primary key for the customer file, allowing quick access to a particular customer record. The field CDESC contains the description of the CI. The two fields FUND and EXPEND, although not used by the system, are included in this file. Those fields provide the flexibility to extend the system and add fund accounting modules.

e. Stockout file

The stock-out file, STOCKOUT.DBF, contains the Stock Out items. The fields SN and CI link the file to the property file and customer file. The VN field defines the transaction.

f. Batch file

The batch file holds all transactions for the day. At the end of the day, transactions are processed and fulfilled based on their priority. As a result of the processing transactions are transferred to either the master or the Stock Out files. The fields are the same as master file.

g. Normal file

The normal file, NORMAL.DBF, contains the values of the normal distribution, which provides the Z value corresponding to sigma. The field EOFZ contains the expected number of stockout with the probability of stockout.

3. INDEX FILES

The proposed system maintains a number of index files, each database file (except normal.dbf) uses at least one index as a primary index for data retrievals during execution of the program. In addition, some files have additional indexes used by programs in printing reports or internal operations.

TABLE 4
DATA DICTIONARY

Field name	Type	Width	Dec	Location
Address	Character	60	0	Customer Address of a customer.
Cdesc	Character	30	0	Customer Common name of the customer. (Ex. The 150 division 1002 regiment)
Class	Character	4	0	Property Material classification. (Ex. 3 for fuel)
Date	Date	8	0	All databases System date, or real date of the record.
Essence	Logic	1	0	Property Combat essential item.
Expend	Numeric	10	2	Customer The amount of \$ spent by customer.
Fund	Numeric	10	2	Customer The allowed fund.
Fundoh	Numeric	10	2	Customer The onhand amount of \$.
Misc	Logic	1	0	Master, Batch Availability to reuse.
NM	Character	16	0	Property Description of the stock number. (Ex. Hammer, ...)
Onhand	Numeric	5	0	Property The amount of the item in storage.
Ost	Numeric	3	0	ASL Order shipping time (lead time to issue).
Outdate	Character	4	0	Stockout Stockout date.
Priority	Character	1	0	Property Issue priority of the customer's.
Qty	Numeric	5	0	Master, Batch, Stockout The quantity involved in the transaction.
RCN	Numeric	4	0	ASL Resource control number. (Ex. 1289 for Rice)
Refilldate	Date	8	0	Stockout Refill date of the stockout item.
ReqNO	Character	16	0	Batch, Stockout, Master Request number composed of zulu date and sequential number of the order.
RO	Numeric	5	0	ASL Requisition objective of a stock item. $RO = SL + EOQ$
ROP	Numeric	4	0	ASL Reorder point of a stock item. $ROP = SL + OST \times \text{demand rate}$
Serialno	Character	10	0	Property Serial number of end item.
SL	Numeric	4	0	ASL Safety level. $SL = OST \times \text{possibility of stockout}$
SN	Character	16	0	Master, ASL, Property, Stockout, Batch Stock number which is specified to every stock item.
Tvalue	Numeric	10	2	Property Total \$ value of onhand quantity.
Type	Character	2	0	Master, Stockout, Batch Transaction type. (Ex. RD = request for issue to division from organizational unit)

TABLE 4
DATA DICTIONARY (CONT'D.)

Unit	Character	4	0	Property Measuring unit of the item. (Ex. Cm, Kg, Each, Drum.....)
Unitcost	Numeric	8	2	Master, Property, Stockout, Batch. Unit price of each item. (Ex. \$20.00 for hammer)
VN	Character	16	0	Master, Batch Voucher number. (Ex. 1500QM-0012-8309)
Zipcode	Character	6	0	Customer Zipcode of a customer's address.

TABLE 5
INDEX FILES

File	Index key	Index name	Description
Property	SN	Snproper	Index by stock number
Customer	CI	Cicust	Index by customer's ID
ASL	SN	Inasl	Index by stock number
Batch	SN+CI	Sncibat	Index by stock number and Customer's ID (Used in transaction issue list)
	CI	Cibat	Index by customer's ID.
Stockout	SN+CI	Stockout	Index by stock number and customer's ID (used in retrieve stockout item which is caused by customer)
Master	SN+CI +TYPE	Sctmast	Index by stock number and Customer's ID and transaction type.
	CI+SN	Csmast	Index by customer ID and stock number

APPENDIX C

SYSTEM STRUCTURE

1. OVERVIEW

The proposed system consists of four modules MANAGFL, TRANSACT, REPORTS and ANALYSIS performing the following four functions file management, transaction processing, report generating and analysis. There are a number of subroutines and performing functions common to one or more modules. The system structure is shown in Figure C.1 The menu system designed for this application corresponds to the gray portion on the figure. Appendix D (User's Manual) contains detailed instructions on how to use the system. This appendix will concentrate on describing the modules.

2. PMMAIN

PMMAIN is the normal starting point of the program and it displays the main menu and stores the last request and voucher numbers issued by the division. The menu is used to select one of the four main modules as shown in Figure C.2

3. MANAGFL

This module is designed for file management and contains ten submodules and several screen formats. During the initial system loading, the user needs to enter data into Property, ASL, and Customer files using the ADDPROP, ADDASL and ADDPROP submodules. During routine operations, the user may correct entered records in Property, ASL, and Customer files by using EDITPROP, EDITASL, and EDITCUST respectively. The user may correct transactions using EDITMAST and EDITBAT.

The flow chart in Figure C.3 illustrates the procedure of ADDPROP.prg. Since the file management procedures are closely related, only one flow chart is presented as an illustration.

The Manageq provides queries on files. The user may consult the present list of stock numbers by stock number, material class, or all stock items. An example is shown on the Figure C.4

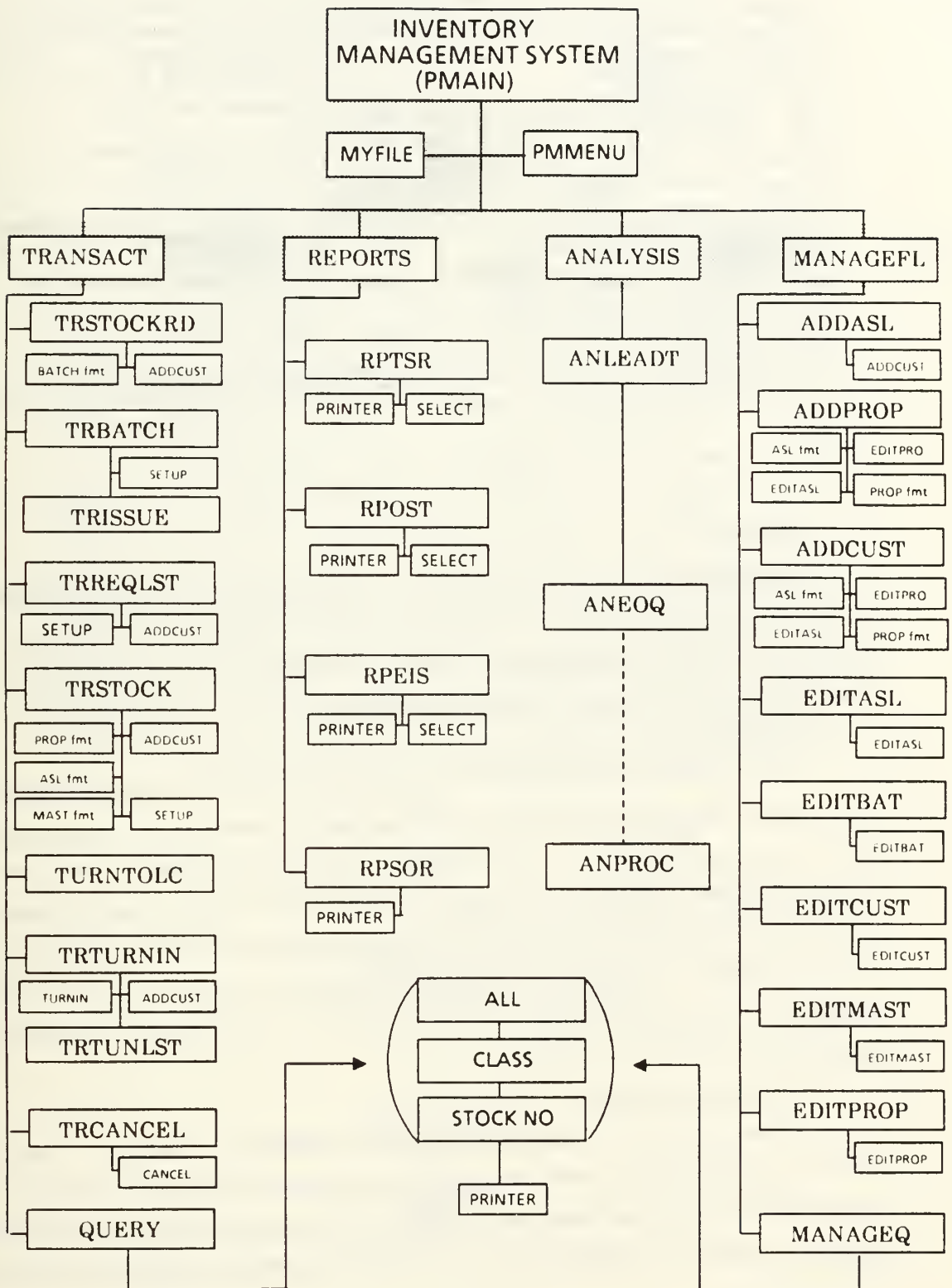


Figure C.1 Detailed system structure

Inventory Management System	
<div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"> 1: Transaction </div> <div style="border: 1px solid black; padding: 5px;"> Routine transaction: Request for issue Receive item Query on transaction </div>	<div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"> 2: Manage file </div> <div style="border: 1px solid black; padding: 5px;"> Management of file: Add Edit (delete, change) Query on files </div>
<div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"> 3: Report </div> <div style="border: 1px solid black; padding: 5px;"> Report Generation: Transaction status report Stockout report Essential item stockout OST report, Fund status </div>	<div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"> 4: Analysis </div> <div style="border: 1px solid black; padding: 5px;"> Analysis of Reorder point Lead time Requisition Objective Safety Level </div>
[Enter Selection (1 - 4, 5 to change date, or 0 to exit: :)]	
Today is 11/18/87 (ZULU:7322)	

Figure C.2 Pmmain screen.

4. TRANSACT

This module includes ten submodules corresponding to the type of transaction, and query on transaction. The overall data flow is shown in the Figure 4.3 This module is the main procedure and contains the analysis and report programs. The main menu for this module is shown in the Figure C.5

a. StockRD

This submodule is designed for transaction type RD - request for issue to the division from the organizational unit. It stores input transactions, e.g. request for issue and turn-in, in a batch file. The submodule checks the validity of both the stock number and the Customer ID. If either is invalid, or the number is not included in the appropriate parent file, i.e. Property and Customer files, the program will prompt the user to update the parent files prior to accepting the entry. In other words no transactions is accepted without a valid SN and CI. The Figure C.6 illustrates the procedure.

b. TRbatch and TRissue

These submodules are the batch file processing procedures. They check whether the ONHAND quantity satisfies the total quantity requested in that day. If the ONHAND is less, transactions are satisfied based on their priority. The submodules generate a request for material when the ONHAND quantity reaches the reorder point.

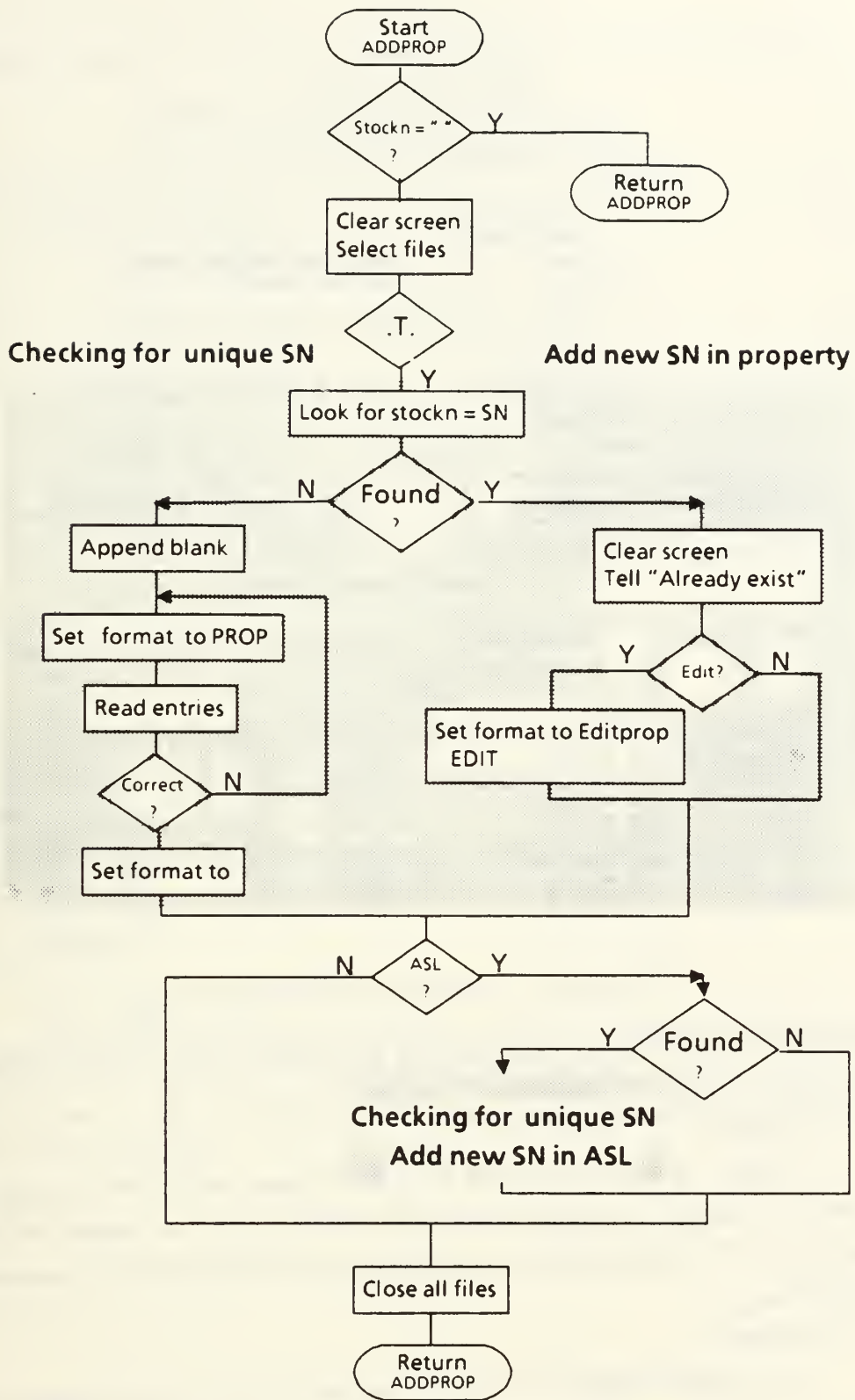


Figure C.3 The flow chart of ADDPROP module

Query on Stock number

Options

1. Stock number
2. Class Enter Class Number: 7
3. All [0] for class 10
4. Return to main menu

Enter Option: 2

Page No. 1
11/16/87

Stock number Listing

Stock number	Nomenclature	Price	Onhand	Requisition Objective	Reorder Potint	Safety Level
** Material Class 7						
7520-00-281-5911	Basket, Waste	10.00	180	180	50	20
7510-00-984-5787	Binder	12.00	93	190	10	3
7920-00-141-5452	Handle, Wood	1122.00	438	10	3	2
7520-00-904-1268	Marker	2.00	200	290	100	30
7510-00-022-8926	Pencil	1.00	520	7000	100	80
7510-00-543-6792	Refill	0.20	1120	1800	900	45
7930-00-141-5888	Wax, Floor	10.00	30	500	25	15
7940-00-168-3366	Widget	120.00	23	0	0	0
7777-77-777-7777	Test item7	26.00	5085	500	280	140

Figure C.4 Example of MANAGEQ module.

Finally they are responsible for aggregating all requests from a single customer into a one issue list. Figure C.7 shows their flowchart.

c. TRreqlst

Module TRREQLIST is used for special events, that is, the manager may be requested to prepare for special events in the division such as training or unplanned construction. This module issues a warning to reflect that the input transaction is not a regular one.

d. TRstock

This module designed for type 'IL' transactions, ie issue from logistics support command. This module validates the customer or supplier ID and the item SN as one of the authorized supplier, initial supplies, and ASL item. A second check determines

Transaction routine	
<div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;">Type of transaction</div> <div style="margin-bottom: 10px;"> 1 : Request for issue to Div 2 : Issue to customer from Div 3 : Request for turn-in to Div 4 : Receive from LSC 5 : Request for issue to LSC 6 : Request for turn-in to LSC 7 : Cancel RL to LSC 8 : Cancel RD to Div 9 : Query on transaction </div>	<div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;">Source</div> <div style="margin-bottom: 10px;"> Stock number : - - - Customer : </div> <div style="border: 1px solid black; padding: 5px;">Information</div> <div style="margin-top: 10px;"> Today is12/02/87 (Zulu : 7336) Last VN : 1500QM-0031-7317 Last Regno : 1500QM-0014-7317 </div>
[Enter Selection from (1 - 9, or 0 to return) :1:]	

Figure C.5 Transaction menu screen.

whether the item received is a stock out item. If it is, the delayed issue list is processed, and all related recording is posted (STOCKOUT.DBF, MASTER.DBF). This process is shown in Figure C.8

e. TURTOLC, TRTURNIN and TRTUNLST

This submodules accepts items returned from units. Turn-in flow chart is shown in Figure C.9

f. Query

This module generates transactions' reprot. Queries such as how many transactions have occurred in given period for specified customer, stock number, material class, or all stock number, may be answered. Figure C.10 shows the screen menu for this submodule.

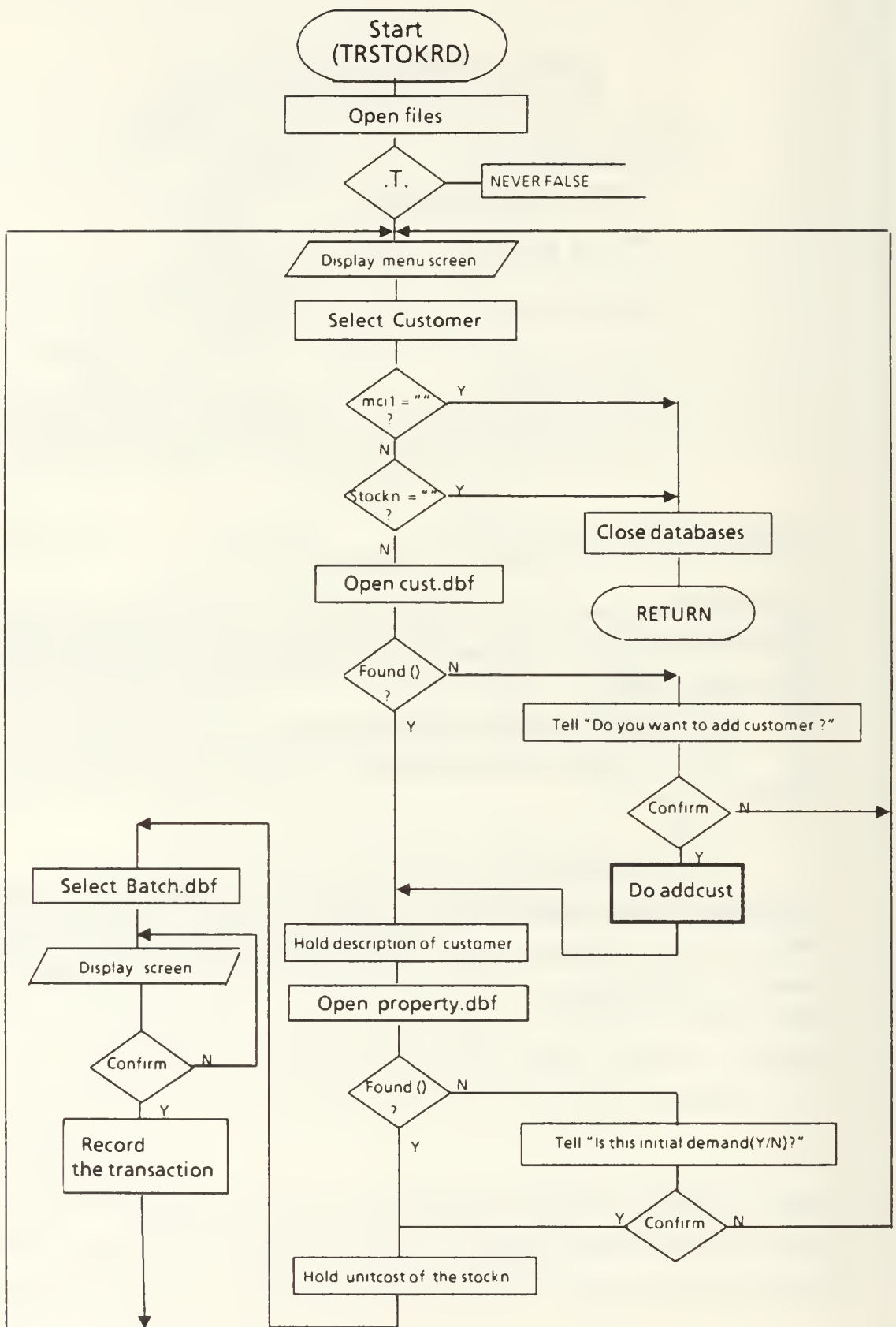


Figure C.6 TRSTOKRD (Request to division) module flow chart

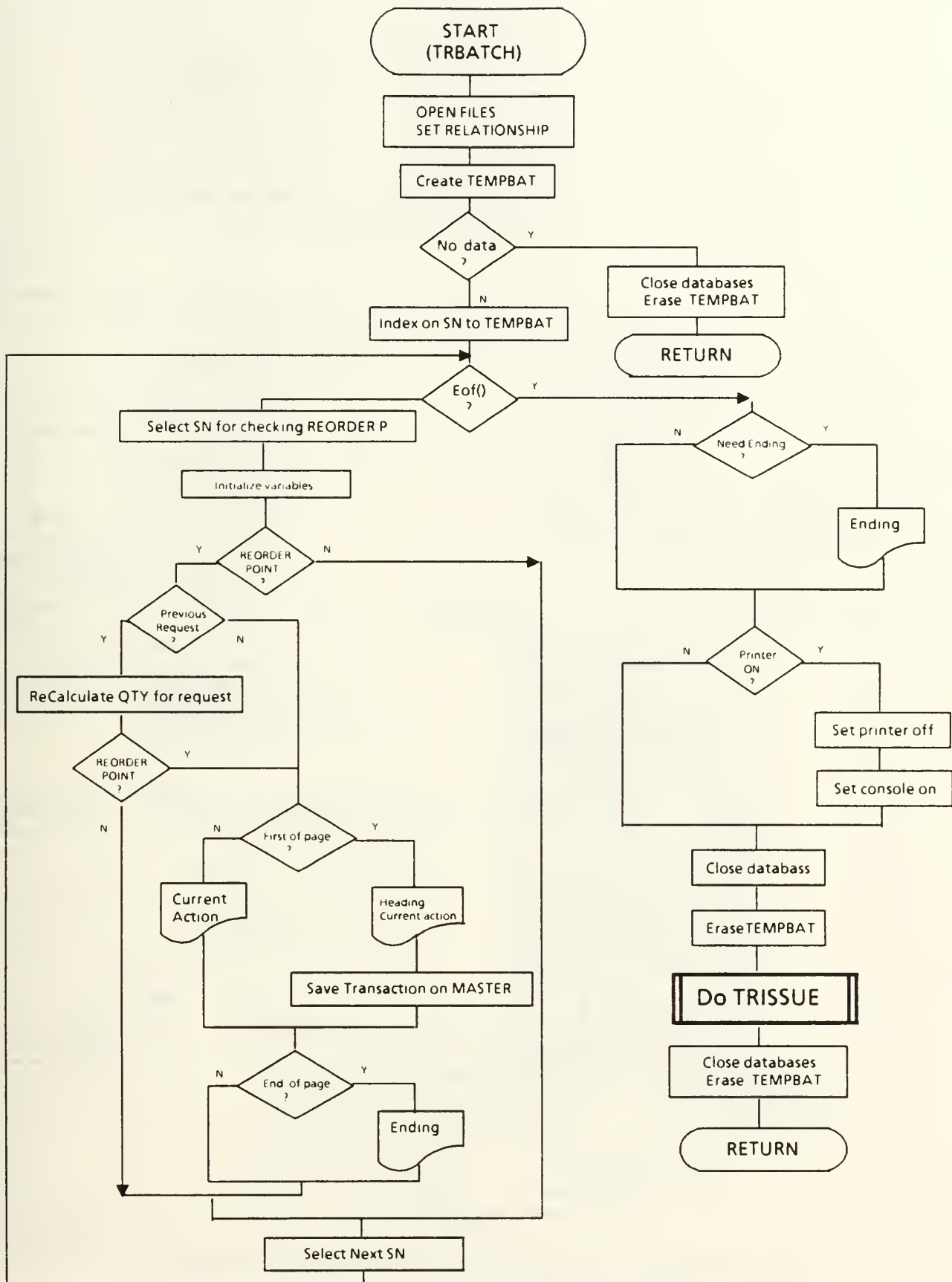


Figure C.7 TRBATCH (batch process) module flow chart

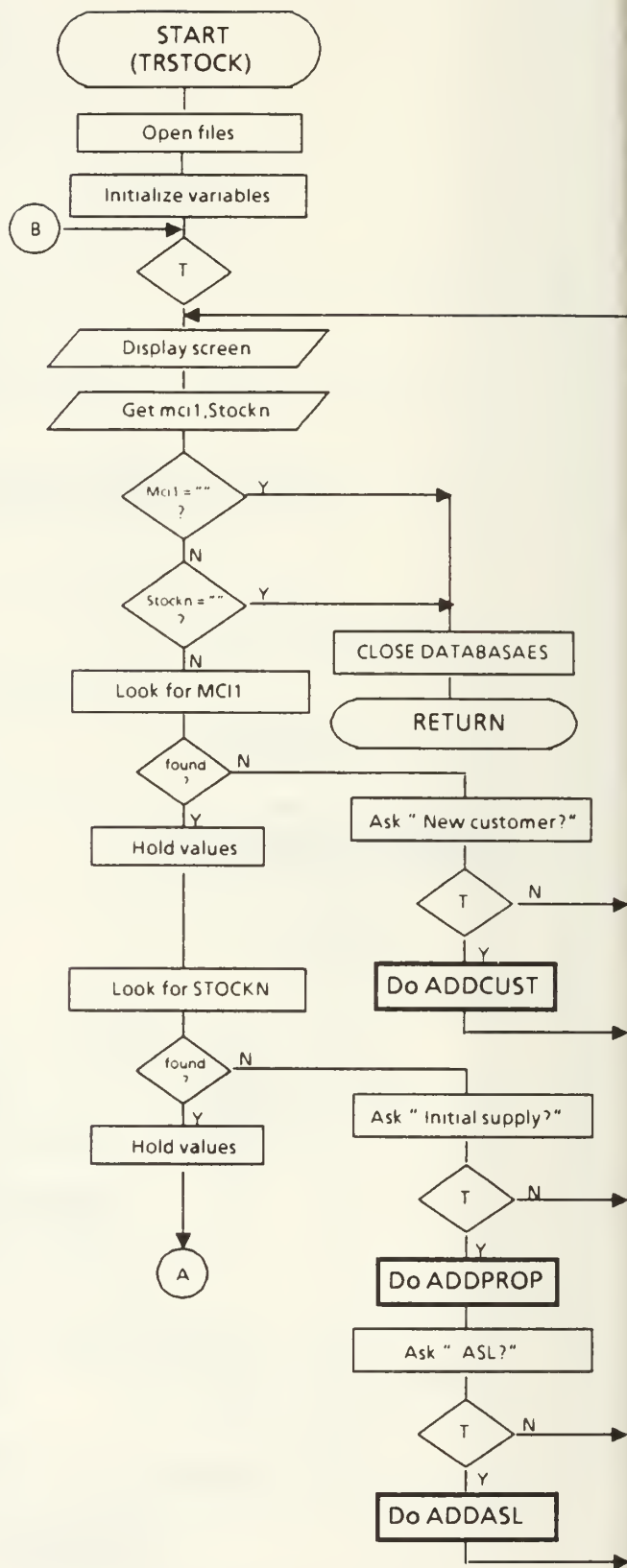
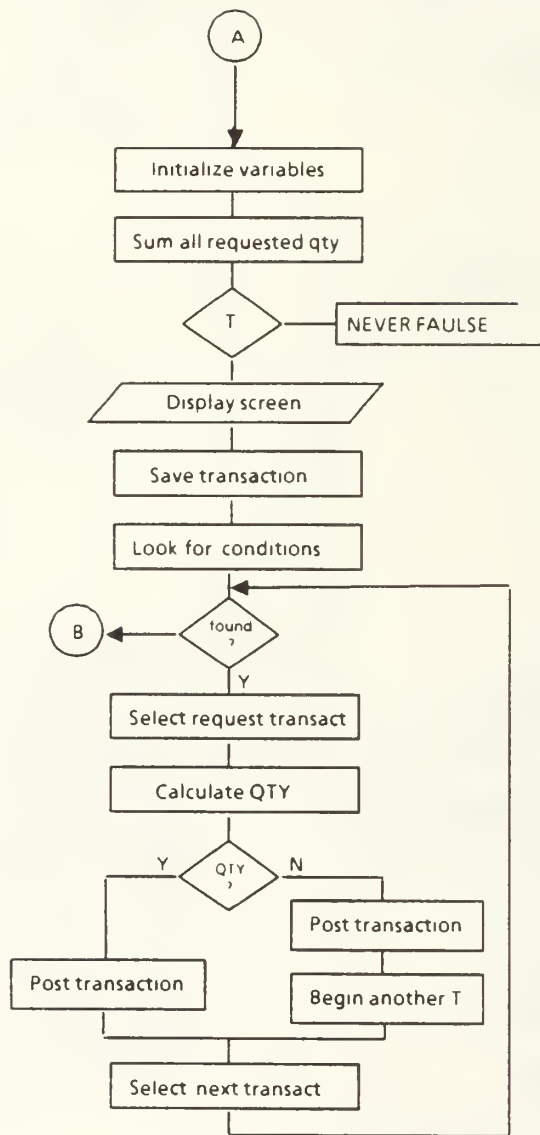


Figure C.8 TRSTOCK (Issue from LSC) module flow chart

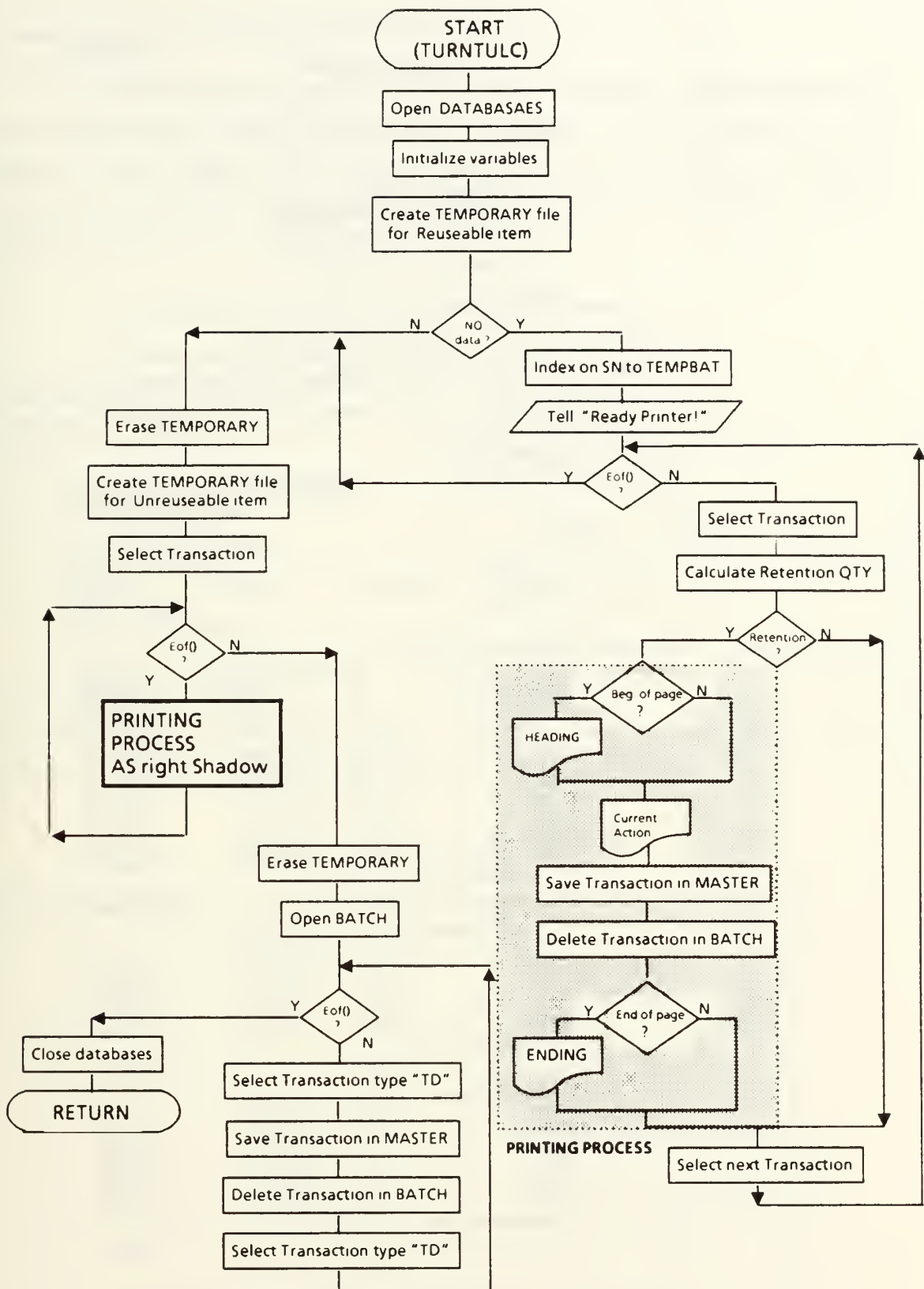


Figure C.9 TURNTULC (Turn-in to LSC) module flow chart

Sales Tracking

Options

1. Detail Sales
2. Summary Sales
3. Detail Customer Sales
4. Summary Customer Sales

5. Return to main menu

Enter Option: 1

Enter period for sales tracking (01/01/87 - 01/01/8)

Page No. 1
11/16/87

Sales Transaction Tracking Detail Sales

Customer ID #	Quantity	Price	Total
** Stock number 1111-11-111-1111			
1500	7	10.00	70.00
1500	88	10.00	880.00
1509	1	10.00	10.00
1509	100	10.00	1000.00
1509	100	10.00	1000.00
1509	10	10.00	100.00
1509	10	10.00	100.00
1509	40	10.00	400.00
3333	95	10.00	950.00
5090	100	10.00	1000.00
5093	100	10.00	1000.00
** Subtotal **			
	651		6510.00
** Stock number 2222-22-222-2222			
1500	97	10.00	970.00
1500	97	10.00	970.00
1504	100	2.00	200.00
1507	10	2.00	20.00
1508	200	2.00	400.00
** Subtotal **			
	504		2560.00
** Stock number 3333-33-333-3333			
1500	25	0.00	0.00
1509	2	102.00	204.00
1509	2	102.00	204.00
5012	2	0.00	0.00
5014	1	102.00	102.00
5032	2	102.00	204.00
** Subtotal **			
	34		714.00

Figure C.10 Example of Query on transaction module.

5. REPORTS

This module consists of RPTSR, RPOST, RPEIS, RPSOR for transaction status, order shipping time, essential item stockout, and stock out reports respectively. Each module creates temporary files and erases them at the end of session. The selection screen in this module is shown on the Figure C.11 followed by examples from the various reprot,

R E P O R T S	
<div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;">Type of Reports</div> <div style="border: 1px solid black; padding: 10px;"><div style="border-bottom: 1px solid black; margin-bottom: 10px;">1 : Transaction Status Report (DA form 3183)</div><div style="border-bottom: 1px solid black; margin-bottom: 10px;">2 : Stockout Report (DA form 3184)</div><div style="border-bottom: 1px solid black; margin-bottom: 10px;">3 : OST Report (DA form 3185)</div><div style="border-bottom: 1px solid black; margin-bottom: 10px;">4 : Combat Essential Item Stockout Report (DA form 3186)</div></div>	<div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;">Information</div> <div style="border: 1px solid black; padding: 10px;"><div style="border-bottom: 1px solid black; margin-bottom: 10px;">Last TSR Report Date 11/12/87</div><div style="border-bottom: 1px solid black; margin-bottom: 10px;">Today....11/16/87(7320)</div></div> <div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;">Time Period</div> <div style="border: 1px solid black; padding: 10px;"><div style="border-bottom: 1px solid black; margin-bottom: 10px;">Beginning Date - Ending Date (/ / - / /)</div></div>
[Enter Selection from (1 - 4, or 0 to return) : :]	

Figure C.11 Report module menu screen.

Rerfer to the Figure C.12, Figure C.13 and Figure C.14

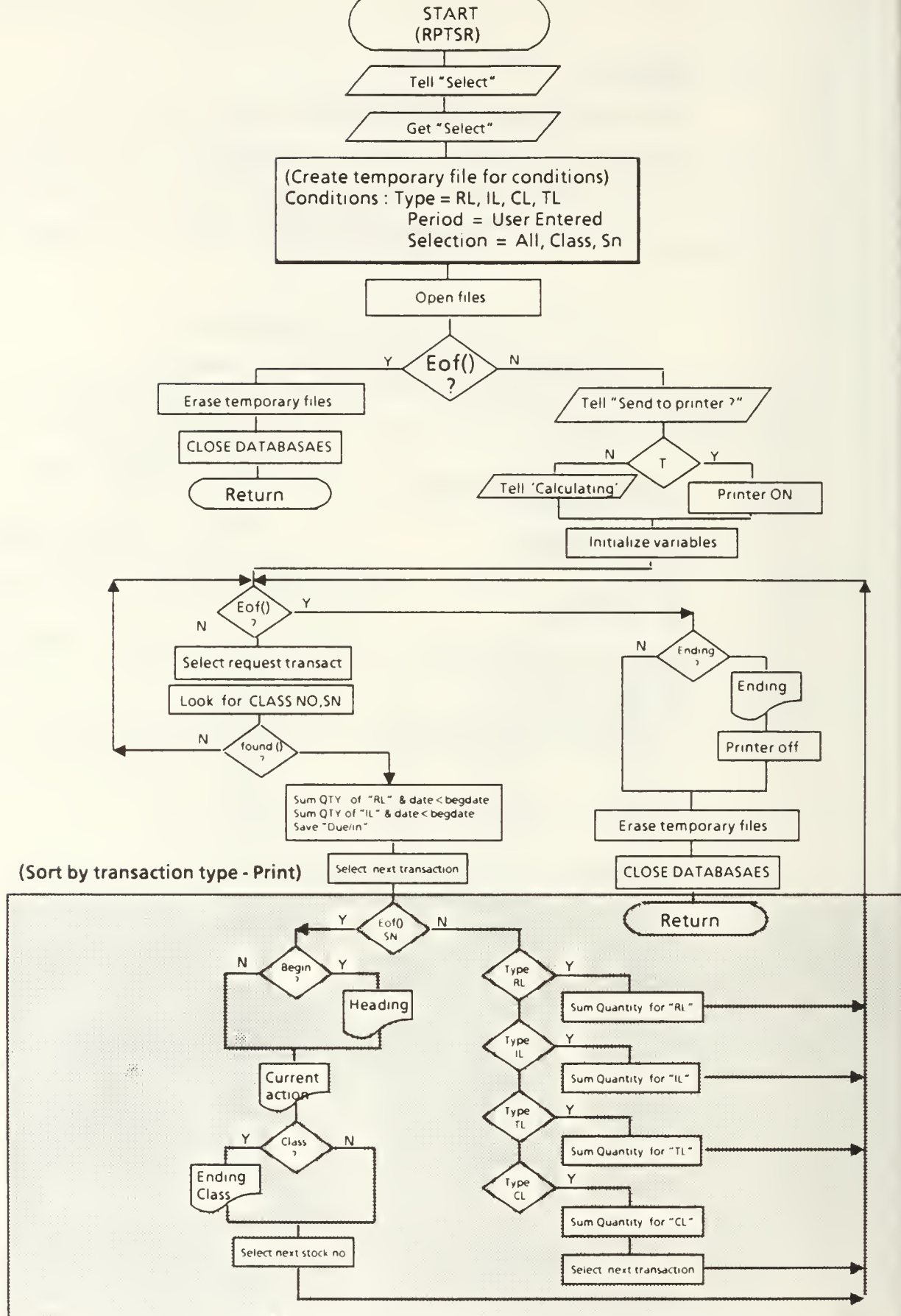


Figure C.12 RPTSR (Transaction status report) module flow chart

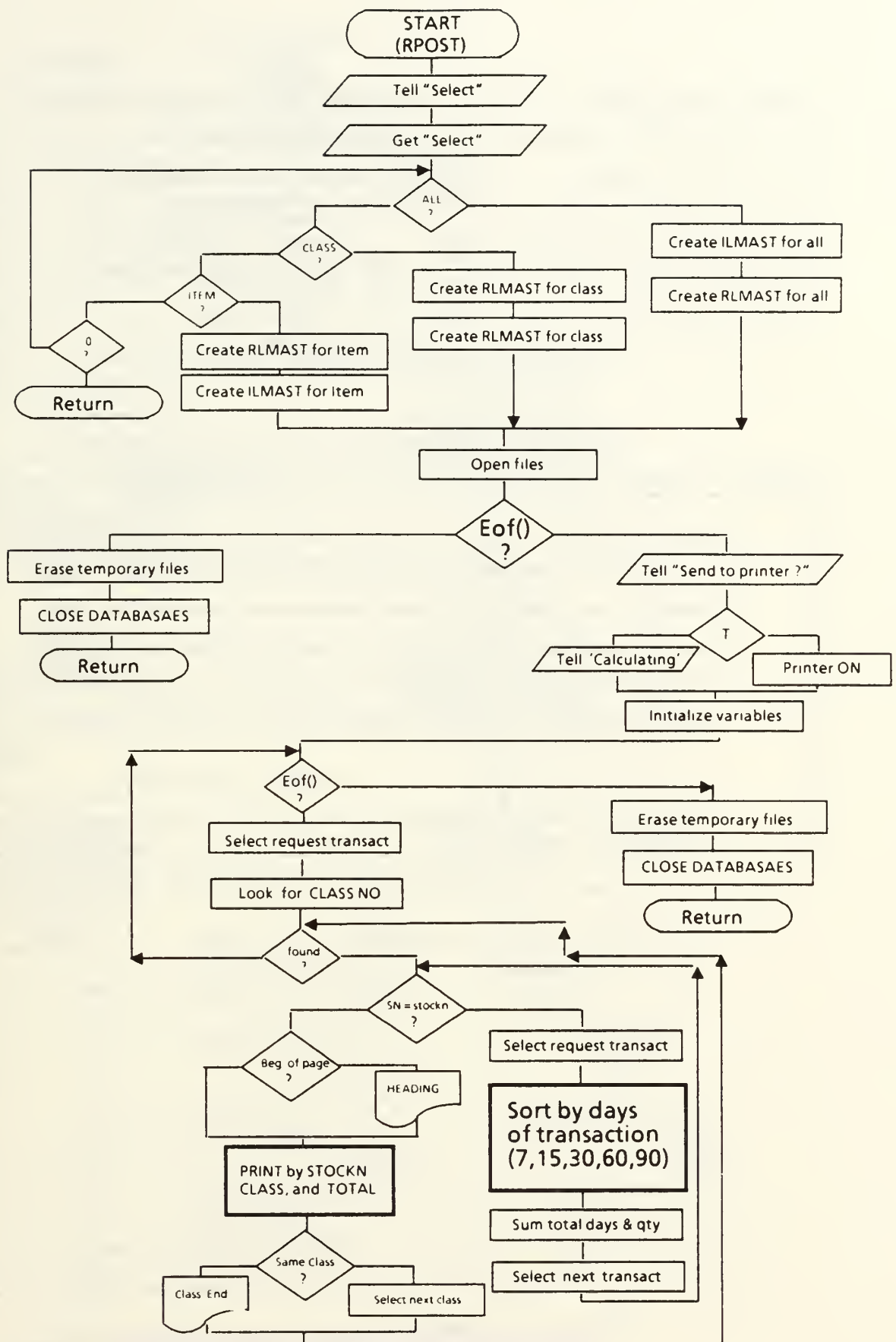


Figure C.13 RPOST (Order shipping time report) module flow chart

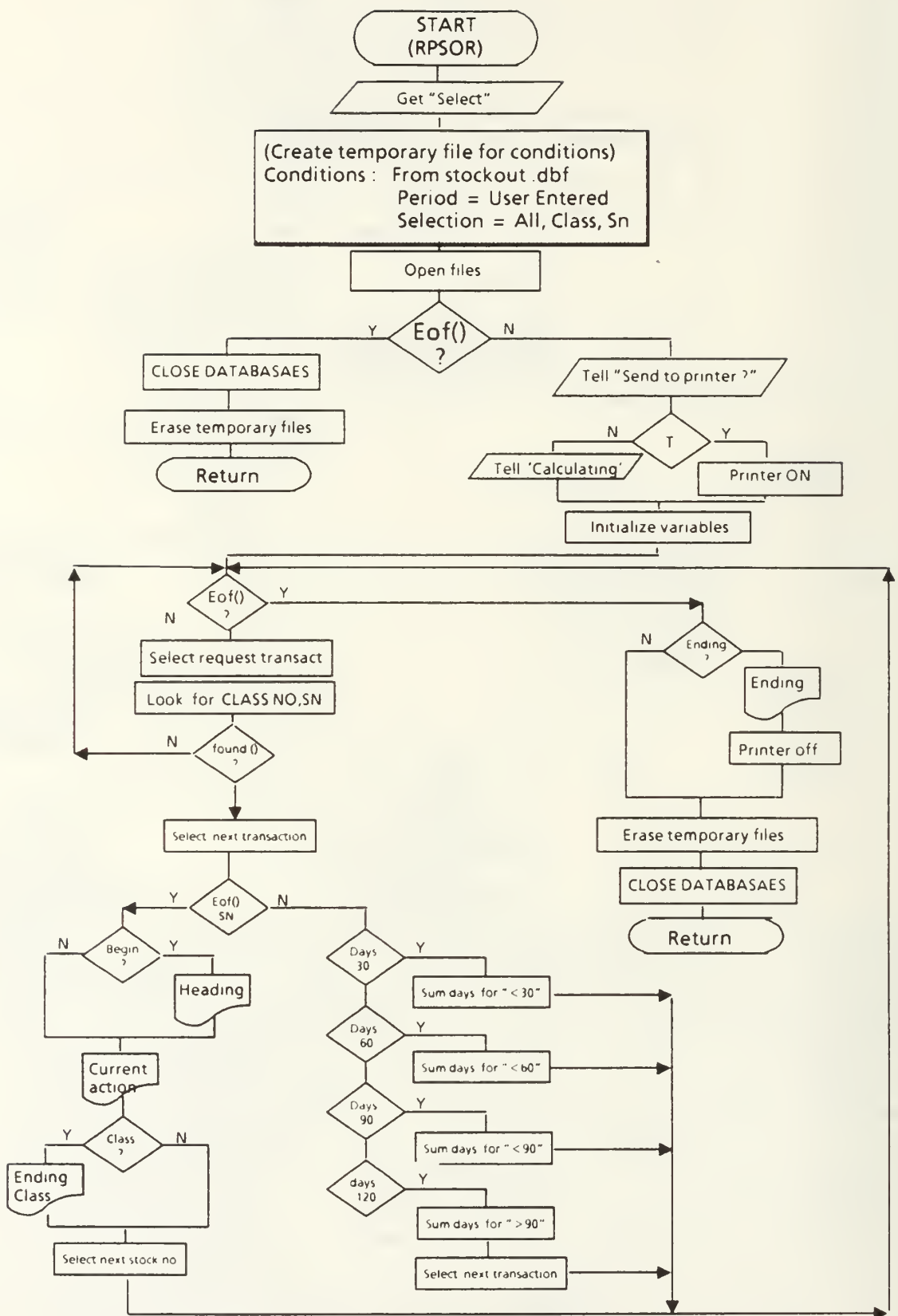


Figure C.14 RPSOR (Stockout report) module flow chart

6. ANALYSIS

This module consists of three procedures, ANLEADT, ANEOQ, and ANPROC and is used to calculate the average lead time of any stock item. The Average Lead Time is used in calculating safety level, reorder point, and finally economic order quantity. The module ANPROC has many procedures which contains zulu date, average, standard deviation, and variance. These procedures are called from the ANALYSIS module whenever needed. The data flow in this module is shown in chapter three limited analysis section. The detailed process is in the flow chart in Figure 4.5 The first screen of this module is shown on the Figure C.15

A N A L Y S I S o f T R A N S A C T I O N		Today : 11/16/87
THIS SYSTEM USES THE FIXED ORDER SIZE SYSTEM WITH PROBABILISTIC MODEL. THE FIXED ORDER SIZE SYSTEM IS COMPLETELY DEFINED BY THE ORDER QUANTITY [Q] AND REORDER POINT [B]. THE RISK OF STOCKOUT OCCURS AFTER REORDER POINT. TO GET [Q],[B], YOU HAVE TO DECIDE SERVICE LEVEL, OR STOCKOUT COST PER UNIT. THIS SYSTEM ASSUMED BACKORDER CASE WITH SERVICE LEVEL INSTEAD OF LOSTSALES CASE.		
Enter stock number : - - -		
Enter time period you want to test(01/01/87 - 12/31/87)		
Begin		End date
Leave BLANK any space to Exit		

Figure C.15 Analysis module screen.

APPENDIX D

USER'S MANUAL

1. INTRODUCTION

The proposed system is for the manager who works at the ROK Army division or lower level logistics management. It is written in dBASEIII Plus programming language. Routine transaction handling, transaction tracking, reports generation, and limited analysis on the transaction records can be accomplished with this system.

The proposed system consists of three diskettes. One contains the software code developed in this thesis, the other two are dBASEIII plus system diskettes.

a. Requirements

The proposed system runs on the IBM PC family of computers, including the XT and AT, along with all true IBM compatibles. DBASEIII PLUS requires the following;

1. *The dBASEIII PLUS program disks and manual.*
2. *An IBM PC, IBM XT, COMPAQ, or other 100 percent IBM PC-compatible computer with a monochrome or color monitor.*
3. *MS-DOS or PC-DOS*
4. *At least 256k of memory. 512k or more is suggested.*
5. *Two 360k floppy disk drives or one 360k drive and a hard disk drive. A hard disk is suggested.*
6. *A printer with at least 80-column capability is suggested.*

[Ref. 8: p. 12]

b. Organization

The remainder of this manual is divided into two sections. Getting Started, describes the contents of the proposed system and how to install it. Working with the Proposed System, describes how to operate the system.

2. GETTING STARTED

The code of proposed system occupies one diskette only. Since the proposed system is not compiled, it needs the dBASEIII plus system diskettes. The proposed system assumes that dBASEIII plus is available in a subdirectory DBASEIII. So install Dbase III Plus in a subdirectory called DBASEIII, see Dbase III manual for instructions.

a. Configuration

Before installing the system, ensure that the config.sys file in your system contains the following commands:

```
BUFFERS = 15
```

```
FILES = 20
```

If your system does not contain a config.sys file, you may copy it from the Dbase III system diskette.

You also need a config.db file in your dBASEIII subdirectory with command "TYPEHEAD = 20". If you do not have this file, copy it from the Dbase III system diskette. If you have config.db add the command and erase config.db file from the proposed system.

you can create either or both files using the DOS editor or the following commands;

```
Copy con config.db
```

```
.....
```

```
TYPEHEAD = 20
```

```
.....
```

You may want to get into the system directly from the bootup. If this is the case, add the following commands to your autoexec.bat file.

```
.....
```

```
PATH = C:\; \DOS; dBASEIII
```

```
CD dBASEIII
```

```
DBASE
```

```
CD \
```

Now you are ready to install the proposed system in your computer.

b. Summary of the proposed system disk.

The thesis system disk contains the following programs and files;

Extention

DBF	ASL	CUSTOMER	NORMAL	MASTER	STOCKOUT	BATCH	PROPERTY
PRG	ANALYSIS	ANLEADT	TRREQSLT	RPTSR	ADDPROP	EDITBAT	MANAGEQ
	SELECT	ANEQ	REPORTS	TRBATCH	ADDASL	TRTURNIN	RPEIS
	PRINT	TRCANCEL	RPOST	RPSOR	TRISSUE	EDITCUST	TRTUNLST
	EDITMAST	CONTINUE	ANPROC	PMAIN	TRANSACTION	TRACKING	EDITPROP
	TURNTOLC	MANAGEFL	PMMENU	TRSTOCK	SETUP	ADDCUST	EDITASL
	TRSTOKRD	PRINTER					

SCR	EDBAT	BATCH	ASL	EDITMAST	EDITPROP	MAST	CUST
	PROP	EDITCUST	EDITASL	TURNIN	CANCMAS		
FMT	PROP	EDITPROP	EDITMAST	EDITCUST	EOQ	EDITBAT	MASKMENU
	TURNIN	BATCH	EDITASL	CUST	MAST	ASL	CANCMAS
FRM	TRANRP1	TRANRP2	TRANRP3	TRANRP4	CUSTRP1	STOCKRP1	
NDX	SNPROPER	CICUST	INASL	CIBAT	STOCKOUT	SCTMAST	SNCIBAT
	CSMAST						

OTHERS INVENTOR.CAT MYFILE.MEM CONFIG.DB CONFIG.SYS If your disk is missing any of previous files, it will not work.

After confirming the existence of all the files, you are ready to install the system in your computer.

It is recommend that you install in a subdirectory called CONTROL in the DBASE III subdirectory. To create the subdirectory, insure that you are in the DBASE III directory and type;

MD CONTROL

Change directory to 'CONTROL' by typing cd dBASEIII\CONTROL and insert the proposed system diskette in drive A, and copy all files into C: using the following commands;

CD dBASEIII\CONTROL

COPY A:*. * C:

CD

Now you are ready to start.

3. WORKING WITH THE PROPOSED SYSTEM

At the DBASE III '.' prompt, type "DO PMAIN" to get the main menu in Figure D.1, the starting point of the system. At the main menu, you may execute one of five actions. Options 1 thru 4 will start one of the four program modules. The fifth, initiated by typing 5, allows you to change the system date.

Inventory Management System	
1: Transaction Routine transaction: Request for issue Receive item Query on transaction	2: Manage file Management of file: Add Edit (delete, change) Query on files
3: Report Report Generation: Transaction status report Stockout report Essential item stockout OST report, Fund status	4: Analysis Analysis of Reorder point Lead time Requisition Objective Safety Level
[Enter Selection (1 - 4, 5 to change date, or 0 to exit: :)]	
Today is 11/18/87 (ZULU:7322)	

Figure D.1 Main menu(PMAIN).

a. Routine transaction

You have reached this point by choosing option 1 from the main menu. This submodule allows to enter transactions and follows the same procedures as the division supply support action among logistics support command and organizational units.

There are nine options on the menu screen, see Figure D.2

1. Request for issue from a customer

This is the procedure to accept a request for issue from your customer and you reach this point by choosing option 1 from the Routine Transaction Menu. This screen is shown in Figure D.3

The program will check the validity of the customer and stock numbers. If either is invalid the program will not accept it and it will prompt you to check its validity or add it to the appropriate file.

Transaction routine	
<div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"> Type of transaction </div> <div style="padding: 5px;"> 1 : Request for issue to Div 2 : Issue to customer from Div 3 : Request for turn-in to Div 4 : Receive from LSC 5 : Request for issue to LSC 6 : Request for turn-in to LSC 7 : Cancel RL to LSC 8 : Cancel RD to Div 9 : Query on transaction </div>	<div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"> Source </div> <div style="padding: 5px;"> Stock number : - - - Customer : </div> <div style="border: 1px solid black; padding: 5px;"> Information </div> <div style="padding: 5px;"> Today is12/02/87 (Zulu : 7336) Last VN : 1500QM-0031-7317 Last Regno : 1500QM-0014-7317 </div>
(Enter Selection from (1 - 9, or 0 to return) :1:)	

Figure D.2 Routine transaction menu screen.

Enter the customer and stock numbers and quantity required. The menu will prompt you for additional items, and the program will generate a request number. For additional customers change the customer's code and follow the same procedure.

2. Issue to the customer

This program will generate an issue list for each customer. You have reached it by choosing option 2 from the transaction routines. This procedure will be executed once every day . The system will integrate all of the request for the day to get the allowance, and consult the availability of the item in stock. If an item is short to meet the requests, it will follow the priority to issue, then report the request for issue to the logistics support command.

You may get a print out if you so desire. Examples of the reports are shown in Figure D.4

3. Request for turn-in

This procedure handles requests for turn-in from a customer. You have reached this option by entering 3 from the Routine Transaction Menu. The screen is shown in Figure D.5, Enter the required elements and, when prompted, prepare the printer. An example of the output is presented below the screen.

Requeset for issue

Today 11/16/87

Requested From : Division aviation bn
Send To : The 150 Infantry Division

Type of transaction : (RD)
Request for Issue to Div

Request No : 1509S4-0019-7320
(Customer's)

Stock Number :5555-55-555-5555

Description :Test item5

Unit :Roll Price : 3.00 \$

Quantity :500

Is this record correct ? X

Figure D.3 Request for issue from customer screen.

4. Receive stock item from LSC

Select option 4 from the Routine Transaction Menu to reach this point. Enter the elements shown on the screen in Figure D.6 The possible output is below the figure.

This procedure is complex and it validates the customer and stock number, checks due-ins and due-outs, and delay issue of the item.

5. Direct request to LSC

Option 5, 6 will work for special request for issue to LSC. This procedure is not for routine transactions. A warning will be issued when you access these options.

6. Cancellation of request

Options 7, 8 will execute user's cancellation requests. Upon selection of one of these options you will receive the screen shown in Figure D.7 After you enter the data, you will be prompted to confirm the record the transaction will be cancel.

Date : 11/13/87

To : The 3333 Logistics Support Command
 From : The 150 Infantry Division

Request No : 1500QM-0014-7317

No	Stock number	Description	Unit	Quantity	Price	Total
1	1111-11-111-1111	Test item1	Ea	105	10	1050
2	8888-88-888-8888	Test item8	Ea	20	22	440

Grand Total :\$ 1490

Matrrial Management NCO : _____ Date : ____/____/____

Material Management Officer : _____ Date : ____/____/____

ISSUE LIST

Page 1

Date : 11/13/87

From : The 150 Infantry Division
 To : The 150 div 1501 regiment

Vocher No:1500QM-0029-7317

No	Stock number	Description	Onhand Unit	Quant'y	Price	Total
1	1111-11-111-1111	Test item1	SORRY! Delay delivery			
2	5555-55-555-5555	Test item5	7976 Roll	300	3	900
3	8888-88-888-8888	Test item8	SORRY! Delay delivery			

Grand Total :\$ 900

Matrrial Management NCO : _____ Date : ____/____/____

Material Management Officer : _____ Date : ____/____/____

Figure D.4 Batch process examples.

b. File management

It is important to enter the Property, ASL and Customer files data first. This will limit the interruptions due to missing CI and SN. To save the necessary data in advance or during the routine transaction job, choose option 2 from the main menu. It will display the screen shown in Figure D.8 From this menu, you can add customer and stock numbers to the Customer, Property, and ASL files. You can also edit records in these files as well as the Master and Batch Files.

There are nine options on this screen, these are

1. Addition

(1) *Customer.* These programs add customer, supplier and item to the appraprate files. On selecting the number 1, the stock number shell will be blocked, and customer shell will await your entry. You must know the customer's code, usually the four digit common name of the unit. After entering the code of the customer, the system validates the record in customer file to prevent duplication. Once confirmed, the next screen which is named 'customer file' in Figure D.9 will appear. Enter the data and confirm that the record is correct. Your confirmation adds the record to the file.

Request for turn-in

Today..... 11/16/87

From : 5010 Regiment 2 Bn
To : The 150 Infantry Division

Type of transaction : (TD)
Request for turn-in

Request No : 5012S4-0011-7320

Stock number : 1111-11-111-1111

Description : Test item1

Unit : Ea

Reuseable? : Y

Price: 10.00 Quantity : 20

Is this record correct ? X

Figure D.5 Request for turn-in from customer.

Receive newstock from LSC

Today is... 11/16/87

Receive from :The 3333 Logistics support Cmd
Send to :The 150 Infantry Division

Type of transaction :
Receive newstock from LSC

Voucher No : 1500QM-0027-7320
Request Number: - -

Stock Number:1111-11-111-1111

Description :Test item1

Unit : EA

Quantity 95 Reusable? : T

Price 10.00

Is this record correct ? X

Figure D.6 Receive stock item from LSC.

To add a property record in your property file, select option two. On selecting the option the customer shell will be blocked and stock number shell will be highlighted. After you have entered the stock number the system will validate your

<div style="border: 1px solid black; padding: 2px; text-align: center;">Cancel Request for issue item</div>	<p>Today ..11/16/87(7320)</p>
<p>Stock number : 2222-22-222-2222</p> <p>Customer Code: 1500</p> <p>Request number: 1500QM-0009-7305</p>	
<p>Quantity : 97 Price : 10.00 Date :(11/0)/87</p>	
<div style="border: 1px solid black; padding: 2px; text-align: center;">Is this record what you want to cancel?(Y/N) N</div>	

Figure D.7 Cancellation of request screen.

Management of files

<div style="border: 1px solid black; padding: 2px; margin-bottom: 10px; text-align: center;">Edit files</div> <ul style="list-style-type: none"> 4.To change Customer (CI) 5.To change Property (SN) 6.To change ASL (SN) 7.To change Master (SN+REQNO) 8.To change Batch (SN+REQNO) 9.Query on files 	<div style="border: 1px solid black; padding: 2px; margin-bottom: 10px; text-align: center;">Addition</div> <ul style="list-style-type: none"> 1.To add Customer (CI) 2.To add Property (SN) 3.To add ASL (SN) <div style="border: 1px solid black; padding: 2px; margin-top: 10px; text-align: center;">Information</div> <p>Today is11/19/87</p> <p>Stock number : XXXX-XX-XXX-XXXX (Or/And)</p> <p>Customer code :</p>
<p>[Enter Selection (1 - 9, or 0 to go to mainmenu) :1:]</p>	

Figure D.8 File management screen.

entry. If accurate, you'll see the next screen, in Figure D.10 Steps similar to customer addition are followed. However, since the ASL list is closely related to the property list,

Customer

File

Today is 11/16/87

Customer Code 2222

Priority 6

Name Test unit

Address Example city

700-11 Zip code

Fund : 0.00 \$ Allowed

0.00 \$ Expenditure

Is this record correct ? Y

Figure D.9 Customer addition screen.

the program will ask you whether the item is also an ASL item and will added to the ASL file if necessary.

Property Book file

Today 11/16/87

Stock number : 8989-89-898-9898

Serial NO :

Description : Example item

Unit : ball Class : 8

On hand : 100

Price : 9.12 \$

Is this combat essential item ? : N

Is this record correct ? : X

Figure D.10 Property list addition screen.

Another method to add an ASL is to add it directly to the ASL file by choosing option 3. It is suggested that you use this method in the case of a newly allowed ASL item. The screen for this option is shown in Figure D.11

Authorized storage list file		Today is 12/02/87
Stock number	:	1111-00-111-0001
Reorder Point	:	100
Safety Level	:	20
Requisition Objective	:	5000
Order Shipping Time	:	12 days
Resource control number	:	1122

Is this record correct ? : X

Figure D.11 ASL list addition screen.

2. Editing or Deletion of an entry

There are five options in this process. Options 4, 5, 6 are equivalent to option 1, 2, 3 respectively. Option 7 allows you to change type TL records, other records' types are automatically saved through the transaction process. Option 8 allows edits of any type of transactions so use it carefully.

(1) *Customer*. From the screen 'management of file' in Figure D.8, select option 4, You will be asked to enter the customer's code that you want to change or delete. On entering the code the program retrieves the customer record from the customer file and displays the next screen in Figure D.12 Edit the screen and exit by pressing the CONTROL and END keys at the same time.

(2) *Property and ASL*. This procedure is almost same as the option 4. Take option 5 for property and 6 for ASL. The edit screen for the ASL list is shown in Figure D.13

Edit or Delete CUSTOMER file		Today 11/16/87
<div style="display: flex; justify-content: space-between;"> Customer : 2222 code Priority : 6 </div> <div style="display: flex; justify-content: space-between; margin-top: 10px;"> <div style="width: 40%;"> Test unit Example city 700-11 </div> <div style="width: 60%;"> Description Address Zipcode </div> </div> <div style="margin-top: 20px;"> Fund allwed : 0.00 0.00 Expend 0.00 On hand </div> <div style="text-align: right; margin-top: 5px;">Last edit date : 11/16/87</div>		
(Insert mode) : Ins (Delete) Character :Del Field: ^Y Record: ^U	(Record) Next : PgDn Previous : PgUp (Done/Save) : ^End Abandon :Esc	

Figure D.12 Customer record editing screen.

Edit Authorized Storage List file		Today 11/29/87
<div style="text-align: center; margin-bottom: 20px;">Stock number : 2222-22-222-2222</div> <div style="display: flex; justify-content: space-between;"> <div style="width: 45%;"> ReOrder Point : 80 Safety Level : 30 Requisition Objective : 180 </div> <div style="width: 50%;"> Order shipping time : 21 days Date : 09/12/87 </div> </div> <div style="text-align: center; margin-top: 20px;">Resource control number : 2222</div>		
(Insert mode) : Ins (Delete) Character :Del Field: ^Y Record: ^U	(Record) Next : PgDn Previous : PgUp (Done/Save) : ^End Abandon :Esc	

Figure D.13 ASL list record editing screen.

(3) *Master file records.* This procedure edits the master file records. It requires you to enter the stock number, the customer's ID, and the date of the transaction. Its screen is shown in Figure D.14

Edit/Change Master file		Today..... 11/29/87
Stock Number : 3333-33-333-3333		
Customer code : 1504	Action Type : RD	
Request Number : 1500QM-0032-7333		
1504S4-0001-7330	Customer's	
Quantity : 10		
Price : 102.\$	Reusable ? : T	
Date : 11/29/87		

(Insert mode) : Ins	(Record) Next : PgDn
(Delete) Character : Del	Previous : PgUp
Field: ^Y Record: ^U	(Done/Save) : ^End Abandon : Esc

Figure D.14 Master file record editing screen.

(4) *BATCH file records.* Sometimes you need to change records already entered in the ASL file through transaction routine. This procedure is more generous than option 7. You may enter the information you have, when you get the screen in Figure D.15, you will find out appropriate record by using the function keys which are given below the screen.

Edit(change or delete) Batch file

Today..... 11/29/87

Stock number : 3333-33-333-3333

Voucher number : 1504S4-0001-7330

Customer Code : 1504

Type of action : RD

Posted :

Quantity : 10

Unitcost : 102.00 \$

Date : 11/29/87

(Insert mode) : Ins

(Record) Next : PgDn

(Delete) Character : Del

Previous : PgUp

Field: ^Y Record: ^U

(Done/Save) : ^End Abandon : Esc

Figure D.15 Batch file record editing screen.

c. Reports generation

Four Types of reports are available in the system. Each report, other than combat essential item stock-out report, may be generated by stock number, by material class, and all. You can get both printed and screen reports. The menu screen is presented in Figure D.16

1. Transaction status report

To get the transaction status report, select option 1 and enter appropriate time period. Select any option you want from the screen in Figure D.17 For example, if you want to get the transaction status reports on material class 2, select option 2. You will be asked to enter the material class, and the type of report, ie screen or print out. It is recommended that you postpone requesting the print out till you check the output on the screen.

Examples of each option are presented in Figure D.18

2. Other Reports

Other reports can be obtained using the same procedure as the transaction status report. Select 2 for stock-out, 3 for order shipping time, 4 for combat essential item stock-out reports.

R E P O R T S	
<div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;">Type of Reports</div> <div style="margin-bottom: 10px;">1 : Transaction Status Report (DA form 3183)</div> <div style="margin-bottom: 10px;">2 : Stockout Report (DA form 3184)</div> <div style="margin-bottom: 10px;">3 : OST Report (DA form 3185)</div> <div>4 : Combat Essential Item Stockout Report (DA form 3186)</div>	<div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;">Information</div> <div style="margin-bottom: 10px;">Last TSR Report Date 11/12/87</div> <div>Today....11/16/87(7320)</div> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;">Time Period</div> <div style="margin-top: 10px;">Beginning Date - Ending Date (. / / - / /)</div>
[Enter Selection from (1 - 4, or 0 to return) : :]	

Figure D.16 Reports generation menu screen.

Sales Tracking

Options

1. Detail Sales
2. Summary Sales
3. Detail Customer Sales
4. Summary Customer Sales
5. Return to main menu

Enter Option: 1

Enter period for sales tracking (01/01/87 - 01/01/8)

Figure D.17 Transaction status report option menu.

d. Analysis

From the main menu screen select option 4. When you get the screen in Figure D.19 enter the stock number you want to analyze and the time period.

From : The 150 Infantry Division
To : The 3333 Logistics Support Command

No	Stock number	Description	From 01/01/87	To 01/01/88	D/Ibeg	Req'st	Cancel	Rec've	Turnin	D/Inow
1	1111-11-111-1111	Test item1			0	950	0	0	0	950
Class 1			-----							
		Subtotal			0	950	0	0	0	950
1	2222-22-222-2222	Test item2			0	970	0	0	0	970
Class 2			-----							
		Subtotal			0	970	0	0	0	970
1	3333-33-333-3333	Test item3			0	0	0	0	0	0
Class 3			-----							
		Subtotal			0	0	0	0	0	0
1	8888-88-888-8888	Test item8			0	0	0	264	0	-264
Class 8			-----							
		Subtotal			0	0	0	264	0	-264
1	9999-99-999-9999	Test item9			0	0	0	24	0	-24
Class 9			-----							
		Subtotal			0	0	0	24	0	-24

Matrerial Management NCO : _____ Date : ____/____/____

Material Management Officer : _____ Date : ____/____/____

Stockout Report

Page 1
Date : 11/06/87

To : The 3333 Logistics Support Command
From : The 150 Infantry Division

No	Stock number	Total		30days		60days		90days		120days	
		Total	Cost	Item	Cost	Item	Cost	Item	Cost	Item	Cost
1	1111-11-111-1111	0	0	0	0	0	0	0	0	0	0
Class 1		-----									
	Subtotal	0	0	0	0	0	0	0	0	0	0
1	2222-22-222-2222	370	740	370	740	0	0	0	0	0	0
Class 2		-----									
	Subtotal	370	740	370	740	0	0	0	0	0	0
1	5555-55-555-5555	1000	3000	1000	3000	0	0	0	0	0	0
Class 5		-----									
	Subtotal	1000	3000	1000	3000	0	0	0	0	0	0
1	8888-88-888-8888	100	2200	100	2200	0	0	0	0	0	0
Class 8		-----									
	Subtotal	100	2200	100	2200	0	0	0	0	0	0

Matrerial Management NCO : _____ Date : ____/____/____

Material Management Officer : _____ Date : ____/____/____

Figure D.18 Examples of Transaction status reports.

The time period will effect the output results. Choose longer period to increase the confidence in the results.

A N A L Y S I S o f T R A N S A C T I O N		Today : 11/16/87
THIS SYSTEM USES THE FIXED ORDER SIZE SYSTEM WITH PROBABILISTIC MODEL. THE FIXED ORDER SIZE SYSTEM IS COMPLETELY DEFINED BY THE ORDER QUANTITY [Q] AND REORDER POINT [B]. THE RISK OF STOCKOUT OCCURS AFTER REORDER POINT. TO GET [Q],[B], YOU HAVE TO DECIDE SERVICE LEVEL, OR STOCKOUT COST PER UNIT. THIS SYSTEM ASSUMED BACKORDER CASE WITH SERVICE LEVEL INSTEAD OF LOSTSALES CASE.		
Enter stock number : - - -		
Enter time period you want to test(01/01/87 - 12/31/87)		
Begin		End date
Leave BLANK any space to Exit		

Figure D.19 Analysis start-up screen.

You may try as many time as you want with different possible data. However you should select one method only. If you do not enter an element, default data will be assumed (ie, service level 85% and holding cost rate 15% of the annual inventory). After entering data into the screen in Figure D.20, the program will display the results and will instruct you on how to save them. The result screen is shown in Figure D.20

I N F O R M A T I O N f o r A N A L Y S I S

You selected stock number : 8883-88-888-3888
(Test item8, Unit:Ea, Class:8)

Purchasing price(P) : 22 S/Ea

Annual Demand(R) : 100 Ea/year

Lead time(OST) in Month : 0 Month(s)

Ordering cost(C) : 0.00 S/order

Holding cost unit per year : 0.00 %
(Select one of these)

Stockout cost(If Known) : 0.00 S/unit Select? N

Service Level in year : 0.0000 % Select? Y

[PRESENT] Reorder point :15

Safety Level :10

Requisition objective :50

Lead Time : 14

Is this record right?(Y/N): X

Figure D.20 Variable entering for analysis.

e. Query

Two types of queries are available. One is transaction tracking, and the other queries on a customer or present status of a stock number.

For the management purposes the transaction records on a stock item or customer's records may be monitored. You may ask for detailed or summarized transaction record. Select option 9 from transaction menu screen. After the screen on Figure D.21 comes up, select any option.

For example:

If you want a customer's transaction detailed record, select option 1. You will be prompted for type of output, ie screen or print out.

If you need to know the present inventory status on a stock number, material class, or all items, select option 9 from the file management menu on Figure D.8 The next procedure is the same as the transaction tracking. The options are presented on Figure D.22

Sales Tracking

Options

1. Detail Sales
2. Summary Sales
3. Detail Customer Sales
4. Summary Customer Sales
5. Return to main menu

Enter Option: 1

Enter period for sales tracking (01/01/87 - 01/01/8)

Figure D.21 Transaction tracking.

Query on Stock number

Options

1. Stock number
2. Class Enter Class Number: 7
3. All [0] for class 10
4. Return to main menu

Enter Option: 2

Figure D.22 Query on stock number.

APPENDIX E

PROGRAM LISTING

1 MAIN MENU

a. PMAIN

```

*****
***** PMAIN.PRG *****
*****
* Module name....: PMAIN.prg *
* Author.....: Park, Taeyong *
* Date.....: Aug 10. 1987 *
* Purpose.....: Main menu of Inventory Management System *
*               for Republic of Korea Army Infantry Division *
* Called by.....: *
* Modules called: PMMENU.prg,MANAGEFL.prg,TRANSACTION.prg *
*               REPORTS.prg,ANALYSIS.prg *
* Variables used.: *
*   Public.: MCI holds customer identification code *
*            STOCKN holds stock number which identifies item *
*            TODAY holds current system date *
*            CHECK holds condition *
*            ZULU holds zulu date which is converted *
*            REQNO1 holds request number which issued the last *
*            VN1 holds voucher number issued the last *
*            MTYPE holds type of transaction *
*            VAR holds the value of variance *
*            STD holds the value of standard deviation *
*            MLEADT holds the value of leadtime of any item *
*   Local...: Z1DAY,Z2DAY,Z3DAY *
*****
* .....Close all open files and clear all variables.
CLEAR ALL
CLOSE ALL
* .....Set working environment.
SET TALK OFF
SET BELL OFF
SET HEADING OFF
SET HELP OFF
SET MENU OFF
SET SAFETY OFF
SET STATUS OFF
* .....
* .....This sets up the CRASH.TXT file which records all .....*
* .....actions so that if the system crashes, the database....*
* .....can be recreated. This file will be deleted if.....*
* .....the system terminate normally.....*
* .....
*SET ALTE to CRASH
*SET ALTE ON
* .....Define the public variables which can be used any module....*
* ..... without redefine.....*
Public MCI,STOCKN,CHECK,ZULU,REQNO1,VN1,MTYPE,VAR,STD,MLEADT
* .....Restore memorized variables from memory file.*
Restore from MYFILE.mem
Store space(4) to MCI,ZULU,MTYPE
STOCKN = space(16)
CHECK = SPACE(1)
* ..... Create memory variable for today's date..*
TODAY = date()
* .....Convert current date into zulu date..*
Set century on

```

```

ZDAY = DTOC(TODAY)
Store "1/1/" + substr(zday,7,4) to Z1DAY
Z2day = TODAY - ctod(Z1day)+1
Store substr(ZDAY,10)+str(Z2DAY,7,3) to Z3DAY
Store substr(Z3DAY,1,4) to ZULU
Set century off
DO WHILE .T.
    CLEAR
    DO PMMENU
    DO WHILE .T.
        i=0
        DO WHILE i=0
            i=INKEY()
            @ 22,64 SAY ""
            IF UPPER(CHR(i))$"012345"
                EXIT
            ENDIF
            i=0
        ENDDO
        @ 22,64 SAY UPPER(CHR(i))
        IF .NOT. CHR(i)$"5"
            EXIT
        ENDIF
        @ 24,38 GET today
        READ
        Set century on
        ZDAY = DTOC(TODAY)
        Store "1/1/" + substr(zday,7,4) to Z1DAY
        Z2day = TODAY - ctod(Z1day)+1
        Store substr(ZDAY,10)+str(Z2DAY,7,3) to Z3DAY
        Store substr(Z3DAY,1,4) to ZULU
        Set century off
        @ 24,38 SAY today
        @ 24,53 SAY ZULU
    ENDDO
DO CASE
    Case CHR(I) = '1'
        DO TRANSACT
    Case CHR(I) = '2'
        DO MANAGEFL
    Case CHR(I) = '3'
        DO REPORTS
    Case CHR(I) = '4'
        DO ANALYSIS
    Case CHR(I) = '0'
        Release All like M*
        Release TODAY,i,CHECK,STOCKN
        Save to MYFILE.mem
        SET TALK ON
        SET BELL ON
        SET HEADING ON
        SET HELP ON
        SET MENU ON
        SET SAFETY ON
        SET STATUS ON
        CLEAR ALL
        EXIT
    OTHERWISE
        ? chr(7)
ENDCASE
ENDDO
SET ALTE OFF
CLOSE ALTE
ERASE CRASH.txt
CLOSE DATABASES
CLEAR

```

```

*.....When done, exit from the system
RETURN
*-----* Eof: PMAIN.prg *-----*

```

b. PMMENU

```

*****
* Module name....: PMMENU.prg
* Author.....: Park, Taeyong
* Date.....: AUG 10, 1987
* Purpose.....: Provide Menu screen for the PMAIN program
* Called by.....: PMAIN.prg
* Module called...: None
* Variable Used...: Today;holds current system date
*****

```

```

@ 1,9 TO 3,69
@ 4,1 TO 23,77 DOUBLE
@ 6,3 TO 12,37
@ 5,4 TO 7,20 DOUBLE
@ 6,5 SAY SPACE(15)
@ 6,41 TO 12,75
@ 5,42 TO 7,59 DOUBLE
@ 6,43 SAY SPACE(16)
@ 14,3 TO 21,37
@ 13,4 TO 15,20 DOUBLE
@ 14,5 SAY SPACE(15)
@ 14,41 TO 21,75
@ 13,42 TO 15,59 DOUBLE
@ 14,43 SAY SPACE(16)
@ 5,2 SAY CHR(176)+CHR(176)
@ 6,2 SAY CHR(176)
@ 7,2 SAY CHR(176)
@ 8,2 SAY CHR(176)
@ 9,2 SAY CHR(176)
@ 10,2 SAY CHR(176)
@ 11,2 SAY CHR(176)
@ 12,2 SAY CHR(176)
@ 13,2 SAY CHR(176)+CHR(176)
@ 14,2 SAY CHR(176)
@ 15,2 SAY CHR(176)
@ 16,2 SAY CHR(176)
@ 17,2 SAY CHR(176)
@ 18,2 SAY CHR(176)
@ 19,2 SAY CHR(176)
@ 20,2 SAY CHR(176)
@ 21,2 SAY CHR(176)
@ 22,2 SAY REPLICATE(CHR(176),75)
@ 21,76 SAY CHR(176)
@ 20,76 SAY CHR(176)
@ 19,76 SAY CHR(176)
@ 18,76 SAY CHR(176)
@ 17,76 SAY CHR(176)
@ 16,76 SAY CHR(176)
@ 15,76 SAY CHR(176)
@ 14,76 SAY CHR(176)
@ 13,60 SAY REPLICATE(CHR(176),17)
@ 12,76 SAY CHR(176)
@ 11,76 SAY CHR(176)
@ 10,76 SAY CHR(176)
@ 9,76 SAY CHR(176)
@ 8,76 SAY CHR(176)
@ 7,76 SAY CHR(176)
@ 6,76 SAY CHR(176)
@ 5,60 SAY REPLICATE(CHR(176),17)
@ 5,21 SAY REPLICATE(CHR(176),21)
@ 6,38 SAY REPLICATE(CHR(176),3)
@ 7,38 SAY REPLICATE(CHR(176),3)

```

```

@ 8,38 SAY REPLICATE(CHR(176),3)
@ 9,38 SAY REPLICATE(CHR(176),3)
@ 10,38 SAY REPLICATE(CHR(176),3)
@ 11,38 SAY REPLICATE(CHR(176),3)
@ 12,38 SAY REPLICATE(CHR(176),3)
@ 13,21 SAY REPLICATE(CHR(176),21)
@ 14,38 SAY REPLICATE(CHR(176),3)
@ 15,38 SAY REPLICATE(CHR(176),3)
@ 16,38 SAY REPLICATE(CHR(176),3)
@ 17,38 SAY REPLICATE(CHR(176),3)
@ 18,38 SAY REPLICATE(CHR(176),3)
@ 19,38 SAY REPLICATE(CHR(176),3)
@ 20,38 SAY REPLICATE(CHR(176),3)
@ 21,38 SAY REPLICATE(CHR(176),3)
@ 2,12 SAY "I n v e n t o r y   M a n a g e m e n t   S y s t e m"
@ 6,6 SAY "1: Transaction"
@ 6,44 SAY "2: Manage file "
@ 14,6 SAY "3: Report "
@ 14,44 SAY "4: Analysis"
@ 8,44 SAY "Management of file:"
@ 9,45 SAY "Add "
@ 10,45 SAY "Edit (delete,change)"
@ 11,45 SAY "Query on files"
@ 8,6 SAY "Routine transaction:"
@ 9,7 SAY "Request for issue "
@ 10,7 SAY "Receive item"
@ 11,7 SAY "Query on transaction"
@ 16,6 SAY "Report Generation:"
@ 17,7 SAY "Transaction status report"
@ 18,7 SAY "Stockout report"
@ 19,7 SAY "Essential item stockout"
@ 20,7 SAY "OST report, Fund status"
@ 16,44 SAY "Analysis of"
@ 17,45 SAY "Reorder point"
@ 18,45 SAY "Lead time"
@ 19,45 SAY "Requisition Objective"
@ 20,45 SAY "Safety Level"
@ 22,8 SAY "Enter Selection (1 - 4, 5 to change date, or 0 to exit: :|"
@ 24,29 SAY "Today is "+"          "+"(ZULU:      ) "
@ 24,38 SAY TODAY
@ 24,53 SAY ZULU
STORE "" TO SELECT
@ 22,63 GET SELECT PICT "9"
RETURN
*-----* Eof: PMMENU.prg *-----*

```

2. ROUTINE TRANSACTION

a. TRANSACTION MENU

```
*****
*****                                TRANSACT.PRG                                *****
*****
* Module name.....: TRANSACT.prg                                           *
* Author.....: Park, Taeyong                                              *
* Date.....: Aug 20. 1987                                                *
* Purpose.....: Record all kind of transaction into the                 *
*               Batch,Master file.                                       *
* Called by.....: PMAIN.prg                                              *
* Modules called : TRBATCH.prg,TRCANCEL.prg,TRSTOCK.prg                  *
*               TRREQSLST.prg,TRSTOKRD.prg,TRTURNIN.prg                  *
*               TRTUNLST.prg,TRTUNLC.prg                                  *
* Variables used...:                                                     *
*               Public...: STOCKN, MTYPE, MCUST, MCI, MREQNO             *
*               Local...: MCI1, MCI2, MHOST, MTYPEA, MTITLE, MISCM, MCDESC1, *
*               MCDESC2                                                  *
*****
```

```
DO WHILE .T.
  Mci1 = space(4)
  Mci2 = space(4)
  Mtitle = space(26)
  Mhost = space(50)
  Mcust = space(50)
  Mtype = space(2)
  Mtypea = space(26)
  Stockn = space(16)
  Mcdesc1 = space(30)
  Mcdesc2 = space(30)
  Mreqno = space(16)
  Miscm = "T"
  CLEAR
  @ 1,15 TO 3,55 double
  @ 4,1 TO 23,77
  @ 6,3 TO 21,38
  @ 5,4 TO 7,28
  @ 6,5 SAY SPACE(23)
  @ 6,41 TO 12,75
  @ 5,42 TO 7,59
  @ 6,43 SAY SPACE(16)
  @ 14,41 TO 21,75
  @ 13,42 TO 15,59
  @ 14,43 SAY SPACE(16)
  @ 2,17 SAY "T r a n s a c t i o n   r o u t i n e"
  @ 6,6 SAY " Type of transaction"
  @ 6,44 SAY "Source"
  @ 14,44 SAY "Information"
  @ 8,6 SAY " 1 : Request for issue to Div"
  @ 9,6 SAY " 2 : Issue to customer from Div"
  @ 11,6 SAY " 3 : Request for turn-in to Div"
  @ 12,6 SAY " 4 : Receive from LSC"
  @ 14,6 SAY " 5 : Request for issue to LSC"
  @ 15,6 SAY " 6 : Request for turn-in to LSC"
  @ 17,6 SAY " 7 : Cancel RL to LSC"
  @ 18,6 SAY " 8 : Cancel RD to Div"
  @ 20,6 SAY " 9 : Query on transaction"
  @ 8, 42 SAY "Stock number : "
  @ 10,42 SAY " Customer : "
  SET COLOR TO N/W
  @ 8,56 SAY STOCKN PICT "9999-99-999-9999"
  @ 10,56 SAY MCI PICT "9999"
  SET COLOR TO
  @ 16,47 SAY "Today is ....."
  @ 16,60 SAY today
  @ 17,51 SAY "(Zulu : "+zulu+)"
  @ 19,47 SAY "Last VN : "+vn1
```



```

@ 20,44 SAY "Last Regno : "+regno1
@ 22,12 SAY "Enter Selection from (1 - 9, or 0 to return) : :"
STORE "" TO SEL
@ 22,59 SAY SEL
*.....The following lines are for select loop.....*
i=0
DO WHILE i=0
  i=INKEY()
  @ 22,59 SAY ""
  IF UPPER(CHR(i))$"0123456789"
    @ 22,59 SAY CHR(i)
    EXIT
  ENDIF
  i=0
ENDDO
DO CASE
  Case CHR(I) = '4'
    MTYPE = "IL"
    Mtitle = "Receive newstock from LSC"
    mhost = "Receive from : "
    Mcust = "Send to : "
    Mtypea = mtitle
    Mci2 = "1500"
    @ 8, 56 GET Stockn pict "9999-99-999-9999"
    @ 10,56 SAY Mci2 pict "XXXX"
    Read
    @ 8, 56 SAY Stockn
    @ 10,56 SAY Mci2
    DO TRSTOCK

  Case CHR(I) = '3'
    MTYPE = "ID"
    mtitle = "Request for turn-in"
    mhost = "Request from : "
    mcust = "Send to : "
    mtypea = mtitle
    mci2 = "1500"
    @ 8, 56 GET Stockn pict "9999-99-999-9999"
    @ 10,56 GET Mcil pict "XXXX"
    Read
    @ 8, 56 SAY Stockn
    @ 10,56 SAY Mcil
    DO TRTURNIN

  Case CHR(I) = '1'
    MTYPE = "RD"
    Mci2 = "1500"
    @ 8, 56 GET Stockn pict "9999-99-999-9999"
    @ 10,56 GET Mcil pict "XXXX"
    Read
    @ 8, 56 SAY Stockn
    @ 10,56 SAY Mcil
    DO TRSTOKRD

  Case CHR(I) = '2'
    MTYPE = "ID"
    @ 8, 56 SAY "XXXX-XX-XXX-XXXX"
    @ 10,56 SAY "XXXX"
    DO TRBATCH

  Case CHR(I) = '5'
    MTYPE = "RL"
    Clear
    @ 10,10 to 17,65
    @ 12,15 SAY "Your selection (RL) must be resulted from other"
    @ 13,15 Say " type of transaction. It means this module "
    @ 14,13 Say "is conducted automatically. However you may"
    @ 15,15 Say " continue this if you eagerly want to do"
    @ 17,15 Say "Do you want to continue(Y/N)? ";
    GET CHECK pict "!"
    Read

```

```

        If CHECK= "Y"
            DO TRREQULST
        Endif
    Loop

    Case CHR(I) = '7'
        MTYPE = "CL"
        @ 8, 56 GET Stockn pict "9999-99-999-9999"
        @ 10,56 GET Mcil pict "XXXX"
        Read
        @ 8, 56 SAY Stockn
        @ 10,56 SAY Mcil
        Do TRCANCEL

    Case CHR(I) = '6'
        MTYPE = "TL"
        @ 8, 56 GET Stockn pict "9999-99-999-9999"
        @ 10,56 SAY Mcil pict "XXXX"
        Read
        @ 8, 56 SAY Stockn
        @ 10,56 SAY Mcil
        Do TURNTOLC

    Case CHR(I) = '8'
        MTYPE = "CD"

    Case CHR(I)= '9'
        MTYPE="QUERY"
        Do Tracking

    CASE I=27 .OR. CHR(I) = '0'
        ? Chr(7)
        CLEAR
        EXIT

    ENDCASE

ENDDO
RETURN
*-----* Eof: Transact.prg *-----*

```

b. TRSTOCKRD

```

*****
*****                                TRSTOCKRD.PRG                                *****
*****
* Module name.....: TRSTOKRD.prg                                              *
* Author.....: Park, Taeyong                                                  *
* Date.....: Aug 28. 1987                                                    *
* Purpose.....: Record request for issue transaction from                    *
*               customer into batch file.                                     *
* Called by.....: TRANSACTION.prg                                             *
* Modules called :                                                            *
* Variables used..:                                                            *
*   Public.:                                                                    *
*   Local...:                                                                    *
*
*****
Select A
Use BATCH index SNCIBAT,CIBAT
Select B
Use PROPERTY index SNPROPER
Select C
Use ASL index INASL
Select D
Use CUSTOMER index CICUST
Select B
Store "X" to batchcheck,checks
Xcheck = .T.
Do while Xcheck
    Clear

```

```

@ 10,15 to 14,65
@ 11,17 Say " Customer Code : " Get Mcil Pict "9999"
@ 13,17 Say " Stock Number : " Get Stockn pict "9999-99-999-9999"
Read
If Mcil = " " .OR. stockn=" "
    Exit
Endif
Select D
Seek mcil
If .NOT. found()
    @ 11,16 clear to 13,64
    @ 11,17 Say "Not found "+Mcil+" code customer "
    @ 13,17 Say "Do you want to add this as customer?(Y/N) ";
    Get checks pict "!"
    Read
    If checks = "Y"
        Mci=Mcil
        Do addcust
        Store cdesc to Mcdesc1
        Loop
    Else
        Loop
    Endif
Else
    Store cdesc to Mcdesc1
    Select B
    StockN = Upper(StockN)
    Seek StockN
    If .NOT. found()
        Clear
        Store "X" to checks
        @ 13,10 say stockn + " is Not found in your property book"
        @ 15,20 Say " Is this the INITIAL demand ? " get checks
        Read
        checks = upper(checks)
        If checks = "Y"
            Store 0 to Mcost
        Else
            Clear
            @ 15, 20 say "Check stock number and try again !"
            Wait
            Loop
        Endif
    Else
        mcost = unitcost
    Endif
Endif
Select A
Append Blank
Replace CI with MCil1, Type with Mtype, SN with StockN,Misc with .T.
Replace Unitcost with Mcost, Date with today, Reqno with Mreqno
Do while .T.
    Set format to batch
    Read
    If batchcheck = "Y"
        Set format to
        Replace REQNO with mreqno
        Stockn = space(16)
        Exit
    Endif
Enddo
Enddo
Close Databases
Release all
Return
*-----* Eof TRSTOKRD.prg *-----*

```

c. TRBATC

```

*****
*****      TRBATC.PRG      *****
*****
* Module name.....: TRBATC.prg
* Author.....: Park,Taeyong
* Date.....: Aug 28. 1987
* Purpose.....: This is the module for performing the
*               BATCH file which contains the records
*               of routine transactions
* Called by.....: TRANSACT.prg
* Modules called : ISSUE.prg
* Databases used.: MASTER,PROPERTY,ASL,BATCH,TEMPBAT
* Variables used.:
*               Public.: REQNO1, ZULU
*               Local..: MQTY holds unit quatity of a given stock item.
*               MTOTAL holds total amount of item which is written
*               on the report so far.
*               MUNITC holds unit price of a given item.
*               STOCKN holds stock number
*               LINECTR holds the number of the line written so far
*               N holds how many iteration has happend
*****

Select A
Use MASTER index SCTMAST
Select B
Use PROPERTY index SNPROPER
Select C
Use ASL index INASL

Select B
Set relation to SN into C
Select A
Set relation to SN into B
Select E
Use Batch index SNCIBAT,CIBAT
*.... Create a temporary file for stock number and type of transaction.*
*.... is same as given condition.....*
Total on SN to TEMPBAT for TYPE="RD"
Use TEMPBAT
Go top

*....If nothing has happened the day, give information and return.....*
If Eof()
  @ 10,14 Clear to 14,65
  @ 10,14 to 14,65
  @ 11,17 Say "Your Batch file is empty -> Go and take rest !"
  @ 13,19 Say "Press Any key to go main menu"
  Wait " "
  Release all
  Close databases
  Erase TEMPBAT.dbf
  Return
Endif

Index on SN to TEMPBAT
Use TEMPBAT index TEMPBAT
Set relation to SN into PROPERTY
Store "X" to mprint,checks
Store 1 to PAGECTR,LINECTR,N
Store 0 to TOTAL,MTOTAL,Z
Go top
Do while .NOT. Eof()
  Clear
  @ 10,14 clear to 14,65
  @ 10,14 to 14,65
  @ 11,27 say "Wait while it is cession "
  Store SN to STOCKN
  Store QTY to NQTY

```

```

Store B->nm to Mnm
Store B->unit to Munit
Store B->unitcost to MUNITC
Store B->ONHAND to MONHAND
Store MONHAND - MQTY to XONHAND
If XONHAND < C->ROP
  *..... Check whether the master file already has an order
  *..... of the item or not, if yes, check the amount so that
  *..... can do action necessary.
  Select A
  Seek STOCKN + "1500RL"
  If found() .AND. .NOT. POSTED
    Store QTY + XONHAND to MONHAND
  Else
    Store XONHAND to MONHAND
  Endif
  Select E
Endif
If MONHAND < C->ROP
  If mprint = "X"
    Do while .T.
      @ 10,14 clear to 14,65
      @ 10,14 to 14,65
      @ 11,27 say "Ready printer? " get MPRINT pict "!"
      Read
      *.Set printer ready to print out, and console off so that
      *......can be being conducted without disturbance.
      If mprint = "Y"
        Set print on
        @ 13,20 Say "Wait while printing....."
        Set console off
      Exit
    Endif
  Enddo
Endif
If LINECTR=40*(N-1)+ 1
  If LINECTR = 1
    Store Val(Substr(REQNO1,8,4)) to MREQNO
    MREQNO = 100000+MREQNO+1
    REQNO1= "1500QM"+"-"+substr(str(MREQNO),7,4)+"-"+ZULU
  Endif
  ? |
  ? |
  ? |
  ? |
  ?? | REQUEST FOR ISSUE'
  ?? | Page ' + str(pagectr,2)
  ?? |
  ? | Date : '+Dtoc(date())
  ? | To : The 3333 Logistics Support Command'+ Mcdesc1
  ? | From : The 150 Infantry Division '
  ? | Request No : '+Reqnol
  ? |
  ? | No Stock number Description '
  ?? 'Unit Quantity Price Total'
  Store pagectr + 1 to pagectr
Endif
Store (C->RO)-MONHAND to MQTY
Store MQTY * MUNITC to MTOTAL
Store TOTAL + MTOTAL to TOTAL
Store linectr to Z
? Str(Z,4)+' '+sn+' '+rtrim(mnm)+space(15 - LEN(rtrim(mnm)))
?? ' '+ munit+' '+str(mqty)+str(munitc)
?? str(munitc*mqty)
linectr = linectr + 1
*..... Record the action on the master file.
Select A
Append Blank
Replace SN with StockN,Ci with "1500"
Replace Reqnol with reqnol,Qty with MQty,Type with "RL"

```



```

Replace Unitcost with munitc,Misc with .T.,Date with today
Replace Posted with .F.
Select E
If linectr=40*N + 1
    ? '
    ?? '
    ? '
    ? '
    ? 'Material Management NCO : '
    ?? 'Date : / / '
    ? 'Material Management Officer : '
    ?? 'Date : / / '
    Eject
    N = N + 1
Endif
Endif
Select E
Skip
Enddo
*..... Print only when heading printed
If linectr > 1
    ? '
    ?? '
    ? '
    ?? '
    ? 'Grand Total :$' + str(total)
    ? '
    ? '
    ? 'Matrial Management NCO : '
    ?? 'Date : / / '
    ? '
    ? 'Material Management Officer : '
    ?? 'Date : / / '
    Eject
Endif
If MPRINT ="Y"
    *..... Reset printer off and console on so that make it
    *..... possible to communicate with the user.
    Set print off
    Set console on
Endif
*..... Erase the temporary file which is no more useable.
Close databases
Erase tempbat.dbf
Do TRISSUE
Release all
Close databases
Return
*-----* Eof BATCH.prg *-----*

```

d. TRISSUE

```

*****
*****          TRISSUE.PRG          *****
*****
* Module name.....: TRISSUE.prg
* Author.....: Park, Taeyong
* Date.....: Sept 2, 1987
* Purpose.....: This is the issue module for generating
*               Receipt to the customer correspondinig
*               to request for issue.
* Called by.....: Batch.prg
* Modules called : none
* Variables used.:
*               Public.: Vn1,
*               Local..: Total,Mtotal,Z,Mexpnd,Mfundoh
* Databases used.: MASTER,PROPERTY,CUSTOMER,BATCH,STOCKOUT
*****

```

```

Select A
Use MASTER index SCTMAST
Select B
Use PROPERTY index SNPROPER
Select C
Use STOCKOUT index STOCKOUT
Select D
Use CUSTOMER index CICUST
Select A
Set relation to SN into B
Select E
Use BATCH index CIBAT,SNCIBAT
Set relation to SN into B
If Eof()
  Clear
  @ 15,25 SAY "Lucky today! Go and take a rest"
  ? Chr(7)
  X=0
  Do while x<35
    X=X+1
  Enddo
  Close databases
  Return
Endif
Store "X" to Mprint
Do while .T.
  @ 10,14 clear to 14,65
  @ 10,14 to 14,65
  @ 11,25 Say 'Ready Printer ? ' Get mprint pict "!"
  Read
  @ 11,42 Say mprint
  If Mprint='Y'
    Set print on
    @ 13,20 Say "Printing....."
    Set console off
    Store 0 to total,mtotal,Z,Mfundoh,Mexpend
    Store 1 to pagectr,linectr,N
    Exit
  Else
    Clear
    Set color to N/W
    @ 13, 20 Say "You Pressed "+mprint
    Set color to
    @ 15, 20 Say "Do you want to finish the job(Y/N)?" Get check
    Read
    If check = 'Y'
      close databases
      Return
    Else
      Store "X" to Mprint
      Loop
    Endif
  Endif
Enddo
Go top
Do while .NOT. eof()
  If delete()
    Skip
    Loop
  Endif
  Store ci to mci
  Select D
  Seek mci
  Store Expend to Mexpend
  Store Fundoh to Mfundoh
  Store cdesc to mcdesc1
  Select E
  Do while Type="RD" .AND. ci = mci

```

```

If delete()
  Skip
  Loop
Endif
Store type to mtype
Store b->unitcost to mcost
Store b->onhand to fonhand
Store qty to Eqty
Store fonhand-Eqty to fonhand
Store reqno to mreqno
Store sn to stockn
If linectr=40*(N-1)+1
  If linectr = 1
    Store Val(Substr(vn1,8,4)) to Mvn
    Mvn=100000+Mvn+1
    Vn1="1500QM"+"-"+Substr(str(Mvn),7,4)+"-"+zulu
  Endif
  ? ' '
  ? ' '
  ? ' '
  ? ' '
  ?? ' '
  ?? ' '
  ?? ' '
  ?? ' '
  ? ' From : '+ 'The 150 Infantry Division'
  ? ' To : '+ mcdesc1
  ? ' '
  ?? ' '
  ? ' No Stock number Description'
  ?? ' Onhand Unit Quant y Price Total'
  store pagectr + 1 to pagectr
Endif
If fonhand < 0
  ? chr(7)
  clear
  @ 15,20 Say "You have stockout item "+stockn
  @ 17,20 Say "This will be added to stockout file"
  num = 0
  Do while num<30
    num=num+1
  Enddo
  Store linectr to Z
  ? Str(Z,2)+' '+sn+' '+rtrim(b->nm)+space(15 -LEN(rtrim(b->nm)))
  ?? ' '+ ' SORRY! Delay delivery'
  Linectr=linectr+1
  Select C
  Append Blank
  Replace sn with stockn,ci with mci,reqno with mreqno
  Replace qty with eqty,unitcost with mcost
  Replace outdate with today,posted with .F.,Type with mtype
  Select E
  Delete
  Loop
Endif
If fonhand >=0
  Store linectr to Z
  ? Str(Z,2)+' '+sn+' '+rtrim(b->nm)+space(15 -LEN(rtrim(b->nm)))
  ?? ' '+str(b->onhand)+' '+ b->unit
  ?? str(qty)+str(b->unitcost)+str(b->unitcost*qty)
  linectr = linectr + 1
  Store b->unitcost*qty to mtotal
  Store total+mtotal to total
  Select A
  Append blank
  Replace sn with stockn,ci with mci,vn with vn1,,
  reqno with mreqno,type with mtype
  Replace qty with eqty
  Replace unitcost with mcost,Misc with .T.;

```

```

        Date with today,Posted with .T.
        Select B
        Seek stockn
        Replace onhand with fonhand,tvalue with fonhand*unitcost
        Select E
        Delete
        Skip
    Endif
    If linectr=40*N+1
        ? '
        ?? '
        ? '
        ?? '          Grand Total ' + str(total)
        ? '
        ? '
        ? '
        ? 'Material Management NCO : 
        ?? 'Date : ____/____/____
        ? 'Material Management Officer : 
        ?? 'Date : ____/____/____
        Eject
        N = N + 1
    Endif
Enddo
If linectr>1 .AND. .NOT. linectr=40N+1
    ? '
    ?? '
    ? '
    ?? '          Grand Total :$' + str(total)
    ? '
    ? '
    ? 'Material Management NCO : 
    ?? 'Date : ____/____/____
    ? 'Material Management Officer : 
    ?? 'Date : ____/____/____
    Store 1 to pagectr,linectr,N
    store 0 to total,mtotal
    Eject
Endif
Select D
Seek mci
Replace Fundoh with Mfundoh-mtotal
Replace Expend with Mexpend+mtotal
Select E
Enddo
Select E
Pack
Store space(4) to mci
Release all
Set print off
Set console on
*-----* Eof TRISSUE.prg *-----*

```

e. TRREQLIST

```

*****
***** TRREQLIST.PRG *****
*****
* Module name.....: TRReq1st.prg *
* Author.....: Park, Taeyong *
* Date.....: Sept 3, 1987 *
* Purpose.....: This is the Transaction module for request *
*               for issue to the higher logistics command *
*               in force. *
* Called by.....: Transact.prg *
* Modules called : None *

```

```
Select A
Use MASTER index SCTMAST
Select B
Use PROPERTY index SNPROPER
Select D
Use CUSTOMER index CICUST

Select A
Store "X" to CHECK
Store 0 to MQTY,TOTAL,MTOTAL
Store Val(Substr(reqno1,8,4)) to Mreqno
Mreqno = 100000+Mreqno+1
Reqno1= "1500QM"+"-"+substr(str(Mreqno),7,4)+"-"+zulu

Do while .T.
  Clear
  @ 10,15 to 14,65
  @ 11,17 Say " Customer Code : " Get Mcil Pict "9999"
  @ 13,17 Say " Stock Number : " Get Stockn pict "9999-99-999-9999"
  @ 15,25 Say "Leave blank any space to exit."
  Read
  Select D
  Seek mcil
  Do Case
    Case Mcil = " "
      Exit
    Case .NOT. found()
      @ 11,16 clear to 13,64
      @ 11,17 Say "Not found "+Mcil+" code customer "
      @ 13,17 Say "Do you want to add this as customer?(Y/N) ";
      Get check pict "!"
      Read
      If check = "Y"
        Do addcust
        Store "X" to CHECK
        Store cdesc to Mcdesc1
      Else
        Loop
      Endif
    Case Found()
      Store cdesc to Mcdesc1
  EndCase
  Select B
  StockN = Upper(StockN)
  Seek StockN
  Do case
    Case stockN = " "
      Exit
    Case .NOT. found()
      Clear
      @ 13,10 say stockn + " is Not found in your property book"
      @ 15,20 Say "Is this the INITIAL demand? " GET CHECK Pict "!"
      Read
      If check = "Y"
        Store 0 to Mcost
        Store "X" to CHECK
      Else
        Clear
        @ 15, 20 say "Check stock number and try again !"
        Wait
        Loop
      Endif
    Case found()
      Store UNITCOST to MCOST
      Store ONHAND to MONHAND
      Store NM to MNM
      Store UNIT to MUNIT
  Endcase
```



```

Select A
Append Blank
Replace CI with MCII, Type with Mtype, SN with StockN, Misc with .T.
Replace Unitcost with Mcost, Date with today, Regno with Regno1
Do while .T.
  Clear
  @ 2, 9 SAY "Request for Issue"
  @ 3, 43 SAY "Today is..."
  @ 3, 56 SAY Today
  @ 5, 7 SAY " To: "+ mcdesc1
  @ 6, 7 SAY "From: The 1500 Infantry Division"
  @ 8, 6 SAY "Type of transaction : "
  @ 9, 6 SAY "Request for Issue(RL)"
  @ 9, 34 SAY "Request Number:"
  @ 9, 50 GET REQNO1 PICTURE "9999!!-9999-9999"
  @ 12, 19 SAY "Stock Number:"
  @ 12, 32 GET MASTER->SN PICTURE "9999-99-999-9999"
  @ 14, 19 SAY "Description : "
  @ 14, 32 GET MNM
  @ 16, 34 SAY "Unit : "
  @ 16, 41 GET MUNIT FUNCTION "!AAA" PICTURE "XXXX"
  @ 18, 19 SAY "Quantity : "
  @ 18, 32 GET MQTY Pict "99999"
  @ 18, 50 SAY "Reusable? : "
  @ 18, 62 GET MASTER->Misc
  @ 20, 19 SAY " Price : "
  @ 20, 32 GET MCOST PICT "999999.99"
  @ 23, 30 SAY "Is this record correct ? " GET CHECK pict "!"
  @ 4, 4 TO 21, 71
  @ 1, 6 TO 3, 40
  @ 7, 5 TO 7, 70
  @ 10, 5 TO 10, 70
  @ 8, 32 TO 9, 32
  Read
  If CHECK = "Y"
    Replace QTY with Mqty
    Store "X" to CHECK
    Exit
  Endif
Enddo
@ 10,12 clear to 14,63
@ 10,12 to 14,63
@ 11,23 say "Ready printer? " GET CHECK pict "!"
Read
If check = "Y"
  Set print on
  @ 13,20 Say "Wait while printing....."
  Store "X" to CHECK
  Set console off
Endif
Store 1 to pagectr, linectr, N
Store 0 to total, mtotal, Z
If linectr <= 1
  ? |
  ? |
  ? |
  ? |
  ?? | REQUEST FOR ISSUE'
  ? | Page ' + str(pagectr,2)
  ? |
  ? | To : '+ Mcdesc1
  ? | From : The 150 Infantry Division '
  ? | Request No : '+Regno1
  ? |
  ? | No Stock number Description '
  ?? 'Unit Quantity price Total'
Endif
Store (Mqty*Mcost) to Mtotal
Store linectr to Z
? Str(Z,4)+' '+sn+' '+rtrim(Mnm)+space(15 - LEN(rtrim(Mnm)))
?? ' '+Munit+' '+str(mqty) + str(Mcost)

```

```

?? str(Mcost*mqty)
Store Linectr+1 to linectr
Store Mcost*qty to mtotal
Store total+mtotal to total
@ 15, 10 clear to 17,65
@ 15,10 to 17,65
@ 16, 15 Say "Do you have more to request(Y/N)? " Get Check Pict "!"
Read
If Check="Y"
    Store "X" to CHECK
    Store 0 to Mqty
    Store space(16) to Stockn
    Loop
Endif
? '
?? '
? '
?? '
? '
Grand Total :$' + str(total)
? '
? '
? 'Matrerial Management NCO : _____
?? ' Date : ____/____/____
? '
? 'Material Management Officer : _____
?? ' Date : ____/____/____
Eject
Store space(16) to Stockn
Enddo
Clear
Set print off
Set console on
Close databases
Release all
Return
*-----* Eof TRREQLST.prg *-----*

```

```

*****
*****          TRSTOCK.PRGM          *****
*****
* Module name....: TRSTOCK .prgm
* Author.....: Park, Taeyong
* Date.....: AUG 18. 1987
* Purpose.....: Record item received from higher command
*               on the master file. If needed add to the
*               ASL or PROPERTY file
* Called by.....: TRANSACT.prg
* Modules called: None
* Variables used.:
*       Public..: Vn1
*       Local..: Docheck,check,fcheck,Dcheck,mqtys,smqtys..
* Databases used.: MASTER,PROPERTY,ASL,CUSTOMER
*****

```

```

Vn1 = "1500QM-"+Substr(str(mvn1),7,4)+"-"+Zulu
Docheck = .T.
Do while DOCHECK
  Clear
  *.....Check Your customer, or add him
  Do while .T.
    @ 10,15 to 14,65
    @ 11,17 Say " Customer Code : " get mcil pict "9999"
    @ 13,17 Say " Stock Number : " get stockn pict;
    "9999-99-999-9999"
    @ 15,20 Say "Leave BLANK stock number to exit"
    Read
    Seek mcil
    If found()
      Mcdesc1 = cdesc
      Mcdesc2 = "The 150 Infantry Division"
      Exit
    Else
      @ 11,16 clear to 13,64
      @ 11,21 Say "Not found "+mcil+" code customer"
      @ 13,21 Say "Do you want to add the customer? " get;
      check pict "!"
      Read
      If check = "Y"
        mci = mcil
        Do addcust
        Store cdesc to mcdesc1
      Endif
    Endif
  Enddo
  *.....Check the new item initial or requested one.
  Select B
  StockN = Upper(StockN)
  Seek StockN
  Do case
    Case StockN = " "
      Docheck = .F.
      Loop
    Case .NOT. found()
      @ 10,15 to 14,65
      @ 11,21 say " Not found in your property book"
      @ 13,21 say " Is this the INITIAL supply?" get;
      check pict "!"
      Read
      If check = "Y"
        Append blank
        Replace SN with StockN,DATE with Today
        Store 0 to unitcost,onhand
        Set format to PROP
        Read
        If fcheck = "Y"
          Set format to
          Store unitcost to mcost
          Store onhand to monhand
          Replace TVALUE with (mcost*monhand)
        Endif
      @ 10,15 clear to 14,65
      @ 10,15 to 14,65
      @ 12,17 Say "Is This "+stockn +;
      " ASL item? " get check pict "!"
      Read
      If check = "Y"
        Select C
        Seek stockN
        If found()
          @ 11,14 clear to 13,64
          @ 11,17 say SN+" is Already exist!"

```

```

@ 13,17 Say "Do you want to ;
change?" get check Pict "!"
Read
If check = "Y"
    Replace SN with Stockn
    Replace Date with Date()
    Do while .T.
        Set format to ASL
        Read
        dcheck= upper(dcheck)
        IF dcheck = "Y"
            Close format
            Exit
        Endif
    Enddo
Endif
Else
    Append blank
    Replace SN with StockN
    Replace date with date()
    Do while .T.
        Set format to ASL
        Read
        If dcheck = "Y"
            Close format
            Exit
        Endif
    Enddo
Endif
@ 10,15 clear to 14,65
@ 10,15 to 14,65
@ 12,25 Say "You did good job. Now try gain!"
@ 13,25 Say " Press Y| to continue " get ;
check Pict "!"
Read
If check = "Y"
    Loop
Endif
Else
    @ 10,15 clear to 12,65
    @ 10,15 to 14,65
    @ 11,25 Say "Sorry no way to go, Try again!"
    @ 13,25 Say " Press Y| to continue " get ;
    Check pict "!"
    Read
    Loop
Endif
Case found()
    mcost = unitcost
    monhand = onhand
Endcase
Select A
Store stockn+"1500"+"RL" to mstockn
Sum QTY for SN=stockn .AND. type="RL" .AND. .NOT. posted to Tqty
Store "X" to zcheck
Do while .T.
    Append Blank
    Replace CI with MCII, Type with Mtype
    Replace SN with StockN,qty with Tqty
    Replace Unitcost with Mcost
    Replace Date with today,misc with .T.,posted with .T.
    Replace vn with vnl
    Set format to mast
    Read
    zcheck = upper(zcheck)
    If zcheck = "Y"

```

```

        Close format
        Store QTY to M1QTY
        Store M1qty to MQTY
        Exit
    Endif
Enddo
Go top
Seek mstockn
*.....If requested item, post it
If found()
    Do while .NOT. Eof() .AND. type="RL" .AND. SN=STOCKN
        If posted
            skip
            loop
        Endif
        Store REQNO to MREQNO
        Store DATE to MDATE
        Store UNITCOST to M1COST
        Store QTY to CQTY
        Do case
            Case M1QTY-CQTY < 0
                Replace vn with vn1,posted with .T.
                Replace QTY with M1QTY
                Append Blank
                Replace sn with stockn
                Replace reqno with mreqno
                Replace date with mdate
                Replace unitcost with m1cost
                Replace qty with cqty-m1qty
                Replace posted with .F.
                Replace ci with "1500",type with "RL"
                Replace misc with .T.
                Exit
            Case M1QTY-Cqty>=0
                Replace vn with vn1,posted with .T.
                Store M1QTY-CQTY to M1QTY
                If M1QTY=0
                    Exit
                Endif
            Endcase
        Skip
    Enddo
    Select B
    Seek STOCKN
    Replace ONHAND with MONHAND+MQTY
    Replace TVALUE with (MONHAND+MQTY)*MCOST
Endif
*..Check If it a stockout refill item.
Select F
Use STOCKOUT index STOCKOUT
Set relation to sn into property
Seek stockn
If Eof() .OR. Bof()
    Stockn = space(16)
    Select D
    Loop
Endif
Store 'X' to mprint
Do while .T.
    @ 10,15 clear to 14,65
    @ 10,15 to 14,65
    @ 11,17 Say " You have a delayed delivery."
    @ 13,17 Say 'Ready Printer ? ' get mprint pict "!"
    Read
    If mprint = 'Y'
        Set print on
        @ 13,17 Say "Printing....."
        Set console off
        Exit
    
```



```

Endif
Enddo
Do while .NOT. Eof() .AND. SN=STOCKN
    If POSTED
        skip
        loop
    Endif
    Store REQNO to MREQNO
    Store CI to MCI
    Store OUTDATE to MDATE
    Select D
    Seek MCI
    Store cdesc to mcdesc1
    Select F
    ? |      |
    ? |      |
    ? |      |
    ? |      |
    ?? |                                     ISSUE LIST'
    ? |                                     Page :1'
    ? | From : '+ 'The 150 Infantry Division'
    ?? |                                     Date : '+DIOC(date())
    ? | To : '+ mcdesc1
    ? |
    ?? | Request Number : '+mreqno
    ? |
    ? | No Stock number Description
    ?? | Onhand Unit Quantity price
    ?? | Total
    ? | 1 '+sn+ ' +rtrim(b->nm)
    ?? space(15 - LEN(rtrim(b->nm)))
    ?? str(b->onhand)+' '+' b->unit
    ?? str(qty) + str(b->unitcost)+str(b->unitcost*qty)
    Store b->unitcost*qty to total
    ? |
    ? |
    ?? | -----
    ? |
    ?? | Grand Total :$'
    ?? str(total)
    ? |
    ? | Matrial Management NCO :
    ?? |
    ?? | Date :_____/_____/_____
    ? |
    ? | Material Management Officer :
    ?? |
    ?? | Date :_____/_____/_____
    Eject
    Store qty to mqtys
    Replace posted with .T.
    Replace refilldate with Today
    Select D
    Seek mci
    Replace expend with expend-total
    Store expend to mexpend
    Replace fundoh with fundoh-mexpend
    Select B
    Replace onhand with monhand -mqtys
    Replace tvalue with (monhand-mqtys)*mcost
    Select A
    Append blank
    Replace SN with STOCKN, CI with MCI, QTY with MQTYS, TYPE with "RD"
    Replace REQNO with MREQNO, UNITCOST with MCOST, POSTED with .T.
    Replace DATE with MDATE, MISC with .T., VN with VN1
    Select F
    Skip

```

```

Enddo
Set print off
Set console on
Stockn = space(16)
Select D
Enddo
Close databases
Clear
*-----* Eof TRSTOCK.prg *-----*

```

g. TRTURN-IN

```

*****
*****          TRTURNIN.PRG          *****
*****
* Program...: Turn-in.prg                      *
* Author...: PARK, Taeyong                    *
* Date....: Sept 2, 1987                      *
* Notes....: Record Request for turn-in from Customer into *
*           Master and property book.         *
*****

Select B
Use PROPERTY index SNPROPER
Select C
Use ASL index INASL
Select D
Use CUSTOMER index CICUST
Select E
Use BATCH index SNCIBAT,CIBAT
Store "The 150 Infantry Division" to Mcdesc2
Store "X" to checks,fcheck,Zcheck
If mci1 = " "
    Close databases
    Return
Endif
Xcheck = .T.
Do while Xcheck
    Store mci1 to custm
    Do while .T.
        Clear
        @ 10,15 clear to 14,65
        @ 10,15 to 14,65
        @ 11,17 Say " Customer Code : " Get mci1 pict "9999"
        @ 13,17 Say " Stock Number : " Get stockn ;
        Pict "9999-99-999-9999"
        @ 15,20 Say "Leave blank Stock Number to Exit"
        Read
        Select D
        Seek mci1
        If .NOT. found()
            @ 16,17 Say "Not found "+mci1+" Code customer...."
            @ 17,17 Say " Please check the code, or add customer first"
            Store "X" to ANS
            @ 16,15 clear to 18,70
            @ 17,17 Say "Do you want Add the customer Now? " ;
            get ans pict "!"
            Read
            If ANS = "Y"
                Store mci1 to mci
                Do ADDCUST
            Endif
        Else
            Store cdesc to mcdesc1
            Exit
        Endif
    Enddo
    If mci1 <> custm

```

```

@ 10,15 clear to 14,65
@ 10,15 to 14,65
@ 12,20 Say "This data will not be added ....."
?? Chr(7)
N = 0
Do while N < 35
    N=N+1
Enddo
Exit
Endif
Select B
StockN = Upper(StockN)
Seek StockN
Do case
    Case stockN = " "
        Select E
        Pack
        Xcheck = .F.
    Case .NOT. found()
        Clear
        @ 13,10 say stockn + " is Not found in your property book"
        @ 15,20 Say "Please check STOCK NUMBER and try again"
        Wait
    Case found()
        mcost = unitcost
        monhand = onhand
        Select E
        Append Blank
        Replace CI with MCII, Type with Mtype, SN with StockN,
        Unitcost with Mcost, Date with Today,Reqno with Mreqno
        Replace POSTED with .F.
        Do while .T.
            Set format to turnin
            Read
            zcheck = upper(zcheck)
            If zcheck = "Y"
                mqty = qty
                Store Reqno to mreqno
                Set format to
                Exit
            Endif
        Enddo
        Clear
        Stockn = space(16)
    Endcase
Enddo
Do TRTUNLST
Close Databases
Release all
Return
*-----* Eof turnin.prg *-----*

```

h. TRTUNLST

```

*****
*****          TRTUNLST.PRG          *****
*****
* Program.: TRTUNLST.prg                *
* Author...: Park, Taeyong              *
* Date....: Sept 2, 1987                *
* Notes...: Report generate program, called from TRTURNIN.prg *
*****
Clear
Select E

```

```

Set relation to sn into property
Store 'X' to mprint
Store 0 to total,mtotal,Z
Do while .T.
  Clear
  @ 10,15 to 15,65
  @ 12,25 Say 'Ready Printer ? ' get mprint pict "!"
  Read
  If mprint = 'Y'
    Set print on
    @ 14,25 Say "Printing....."
    Set console off
    Exit
  Else
    @ 15, 15 Say "Be ready printer ! "
    Wait
  Endif
Enddo
Store Val(substr(Vn1,8,4)) to mvn
Mvn = 100000+mvn+1
Vn1 = "1500QM-"+substr(str(mvn),7,4)+"-"+Zulu
Store 1 to pagectr,linectr,N
Go top
Do while .NOT. eof()
  If type = mtype .AND. ci=Custm
    If .NOT. posted
      If linectr=40*(N - 1) + 1
        Mvn = Val(Substr(Vn1,8,4))
        Mvn = 10000+mvn+1
        Vn1 = "1500QM-"+Substr(str(Mvn),7,4)+"-"+Zulu
        ? |
        ? |
        ? |
        ? |
        ? |
        ?? ' (Turn-In)
        ? |
        ? | From : The 150 Infantry Division '
        ?? |
        ?? | Date : '+DIOC(date())
        ? | To : '+mcdescl
        ? |
        ?? |
        ? | Voucher No : ' +Vn1
        ? |
        ? | No Stock number Description
        ?? | Quantity Unit price Total Reuasble'
        Store pagectr + 1 to pagectr
      Endif
      Store linectr to Z
      If misc
        ? Str(Z,4)+' '+sn+' '+rtrim(b->nm)+space(15-LEN(rtrim(b->nm)))
        ?? '+' str(qty)+''+ b->unit
        ?? str(b->unitcost) + str(b->unitcost*qty)+' '
        ?? Misc
        Store b->unitcost*qty to mtotal
        Store total+mtotal to total
        Replace posted with .T.
        Store B->onhand - qty to monhand
        Select B
        Replace onhand with monhand
        Replace tvalue with monhand*unitcost
        Select E
      Else
        ? Str(Z,4)+' '+SN+' '+rtrim(b->nm)+;
        space(15-LEN(rtrim(b->nm)))
        ?? '+'str(qty)+''+b->unit+'
        ?? misc
        Replace posted with .T.
        Store total to total
      Endif
    Endif
  Endif

```

```

Replace vn with vn1
If linectr=40*N
  ? '
  ?? '
  ? '
  ?? 'Grand Total ' + str(total)
  ? '
  ? '
  ? '
  ? 'Material Management NCO : '
  ?? 'Date : / / '
  ? 'Material Management Officer : '
  ?? 'Date : / / '
  Eject
  N = N + 1
Endif
linectr = linectr + 1
Endif
Skip
Enddo
? '
?? '
? '
?? 'Grand Total :$' + str(total)
? '
? '
? 'Authorized to turn-in
? 'by
? 'Matrrial Management NCO : '
?? 'Date : / / '
? '
? 'Material Management Officer : '
?? 'Date : / / '
linectr = 1
pagectr = 1
Store 0 to total
Eject
Set print off
Set console on
Return
*-----* Eof TRTUNLC.prg *-----*

```

i. TRCANCEL

```

*****
***** TRCANCEL.PRG *****
*****
* Module name.....: TRCancel.prg *
* Author.....: Park, Taeyong *
* Date.....: Sept 20, 1987 *
* Purpose.....: Cancel the request for issue item *
* Called by.....: Transact.prg *
* Modules called : None *
*****

```

```

Use Master index SCTMAST
Store "N" to Checks
Store Stockn to Msn
Store "1500" to Mcust
Store Msn+Mcust+"RL" to Msncu
Seek Msncu
Do case
  Case STOCKN = " "
    More = .F.
    ?? Chr(7)
    @ 11,15 clear to 15,65

```



```

        @ 11,15 to 15,65
        @ 13,17 Say "Please Enter Stock number you want to cancel ! "
        N = 0
        Do while N < 35
            N = N+1
        Enddo
    Case found()
        Set format to CANCMAS
        Read
        If checks= "Y"
            Replace Vn with "CANCEL-0000-0000",posted with .T.
            Store unitcost to mcost
            Store regno to mregno
            Store qty to mqty
            Append Blank
            Replace sn with stockn,ci with "1500",Regno with mregno
            Replace Qty with Mqty,Type with "CL",unitcost with mcost
            Replace Misc with .T.,posted with .T.,Date with today
            Replace Vn with "Cancel-0000-0000"
            More = .F.
        Endif
        Set format to
    Case .NOT. found()
        @ 11,15 clear to 15,65
        @ 11,15 to 15,65
        @ 13,18 Say "You never request "+stockn+" code item !"
        ?? chr(7)
        N=1
        Do while N <35
            N=N+1
        Enddo
        More = .F.
Endcase
Close Databases
Return
*-----* Eof Cancel.prg *-----*

```

j. TRACKING

```

*****
***** TRACKING.PRG *****
*****
* Module name.....: Trancking.prg *
* Author.....: Park, Taeyong *
* Date.....: Sept 20. 1987 *
* Purpose.....: Tracking the record of transaction. *
* Called by.....: Transact.prg *
* Modules called : None *
*****
Set talk off
Set echo off
Use master
Index on Ci+SN to CSMAS
Use master index sctmast,CSMAS
title = "Sales Tracking"
Mok="Y"
Do while .T.
    Clear
    @ 4,25 say title
    @ 8,15 Say "Options"
    @ 10,17 Say "1. Detail Sales"
    @ 11,17 Say "2. Summary Sales"
    @ 12,17 Say "3. Detail Customer Sales"
    @ 13,17 Say "4. Summary Customer Sales"
    @ 15,17 Say "5. Return to main menu"
    @ 17,15 Say "Enter Option: "

```

```

@ 2,1 to 23,75
Store "" to sel
@ 17,29 Get sel
i=0
Do while i=0
    i=inkey()
    If chr(i)$"12345"
        Exit
    Endif
    i=0
Enddo
@ 17,29 Say Chr(i)
If chr(i)="5"
    Exit
Endif
Store space(8) to Enddate
Store Space(8) to begdate
@ 21,9 Say "Enter period for sales tracking (      )"
@ 21,42 Get begdate pict "99/99/99"
@ 21,53 Get Enddate pict "99/99/99"
Read
Store Ctod(begdate) to begdate
Store Ctod(Enddate) to enddate
Set filter to Date>begdate .AND. Date<Enddate
Do case
    Case chr(i)="1"
        Set Order to 1
        Do setup
        Report form tranrp1
    Case chr(i)="2"
        Set Order to 1
        Do setup
        Report form tranrp2
    Case chr(i)="3"
        Set Order to 2
        Do setup
        Report form tranrp3
    Case chr(i)="4"
        Set Order to 2
        Do setup
        Report form tranrp4
    Otherwise
        Loop
Endcase
If Upper(Mok)="Y"
    Set console on
    Set print off
    Set Order to 1
Else
    @ 24,17 Say "Press any key to continue..."
    Wait""
Endif
Clear
Enddo
Close databases
Erase cimast.ndx
clear
Return
*-----* Eof TRacking.prg *-----*

```

3. REPORTS

a. REPORTS MENU

```
*****
*****      REPORTS.PRG      *****
*****
* Module name.....: Reports.prg
* Author.....: Park, Taeyong
* Date.....: Sept 15. 1987
* Purpose.....: To generate report to higher command
*               or to use in division
*
* Called by.....: Pmain.prg
* Modules called : RPTSR.prg RPSOR.prg RPOST.prg RPEIS.prg
* Variables used.:
*               Public.: Reportd(last report date)
*               Local..: Bigdate,Enddate,Msel
*
*****
```

DO WHILE .T.

Store space(8) to begdate,enddate
Store "X" to Msel

CLEAR

@ 1,15 TO 3,55 double

@ 4,1 TO 23,77

@ 6,3 TO 21,38

@ 5,4 TO 7,28

@ 6,5 SAY SPACE(23)

@ 6,41 TO 12,75

@ 5,42 TO 7,59

@ 6,43 SAY SPACE(16)

@ 14,41 TO 21,75

@ 13,42 TO 15,59

@ 14,43 SAY SPACE(16)

@ 2,25 SAY "R E P O R T S"

@ 6,6 SAY "Type of Reports"

@ 6,44 SAY "Information"

@ 14,44 SAY "Time Period"

@ 9,6 SAY "1 : Transaction Status Report"

@ 10,10 SAY "(DA form 3183)"

@ 12,6 SAY "2 : Stockout Report"

@ 13,10 SAY "(DA form 3184)"

@ 15,6 SAY "3 : OST Report"

@ 16,10 SAY "(DA form 3185)"

@ 18,6 SAY "4 : Combat Essential Item"

@ 19,10 SAY "Stockout Report"

@ 20,10 SAY "(DA form 3186)"

@ 8, 46 SAY "Last TSR Report Date"

@ 9, 50 SAY TSRDATE

@ 11,46 SAY "Today...." +DTC(TODAY)+"("+zulu+")"

@ 17,45 SAY "Beginning Date - Ending Date"

@ 19,50 Say "(-)"

SET COLOR TO N/W

@ 19,51 SAY Begdate Pict "99/99/99"

@ 19,62 SAY Enddate Pict "99/99/99"

SET COLOR TO

@ 22,13 SAY "Enter Selection from (1 - 4, or 0 to return) : |"

STORE "" TO SEL

@ 22,60 SAY SEL

*.....The following lines are for select loop

i=0

DO WHILE i=0

i=INKEY()

@ 22,60 SAY ""

IF UPPER(CHR(i))\$"01234"

@ 22,60 SAY CHR(I)

EXIT

ENDIF

```

        i=0
ENDDO
If CHR(I) = '0'.OR. I=27
    ? Chr(7)
    Clear
    Exit
ENDIF
DO CASE
    Case CHR(I) = '2'
        @ 8, 46 SAY "Last SOR Report Date"
        @ 9, 50 SAY SORDATE
    Case CHR(I) = '3'
        @ 8, 46 SAY "Last OST Report Date"
        @ 9, 50 SAY OSTDATE
    Case CHR(I) = '4'
        @ 8, 46 SAY "Last EIS Report Date"
        @ 9, 50 SAY EISDATE
ENDCASE
@ 19,51 GET Begdate Pict "99/99/99"
@ 19,62 GET Enddate Pict "99/99/99"
READ
IF BEGDATE =" " .OR. ENDDATE=" "
    ? CHR(7)
    @ 24,15 SAY "YOU HAVE TO ENTER THE PERIOD"
    K=0
    DO WHILE K<40
        K=K+1
    ENDDO
    LOOP
ENDIF
Store Ctod(begdate) to begdate
Store Ctod(enddate) to enddate
DO CASE
    Case CHR(I) = '1'
        Do RPTSR
        STORE TODAY TO TSRDATE
    Case CHR(I) = '2'
        DO RPSOR
        STORE TODAY TO SORDATE
    Case CHR(I) = '3'
        DO RPOST
        STORE TODAY TO OSTDATE
    Case CHR(I) = '4'
        DO RPEIS
        STORE TODAY to EISDATE
    OTHERWISE
        ? CHR(7)
ENDCASE
ENDDO
RETURN
*-----* Eof: REPORTS.prg *-----*

```

b. RPTSR

```

*****
***** RPTSR.PRG *****
*****
* Module name.....: RPTSR.prg *
* Author.....: Park, Taeyong *
* Date.....: Sept 15. 1987 *
* Purpose.....: Generate Transaction history report *
*               to higher command. *
* Called by.....: REPORTS.prg *
* Modules called : *
* Variables used...: *
*               Local...: Begdate, Enddate, Classn, Type *
*

```

```

Clear
Select B
Use PROPERTY index snproper
Do while .T.
    Store " " to Classn, Mprint
    Store " Transaction Status Reports" to Mtitle
    Do Select
    Store "(Date>=Begdate .AND. Date<=Enddate)" to period
    Store "Type='RL'.OR.TYPE='IL'.OR.TYPE='CL'.OR.TYPE='TL'" to tcond
    Store "&PERIOD" TO CN1
    Select A
    Use MASTER index sctmast
    Copy to Mlmast for &tcond
    Do case
        Case upper(Chr(i))="1"
            Sel="1"
            Select B
            Index on Class+Sn to CSPROPER
            Select A
            Use Mlmast
            Copy to STMAST for &cn1
            Exit
        Case upper(chr(i))="2"
            Sel="2"
            @ 14,30 Say "Enter Class number(1-10) " get Classn pict "9"
            @ 15,30 Say "          0 for 10 "
            Read
            If classn = "0"
                Store "10" to Classn
            Endif
            Select A
            Use MlMAST
            Copy to XTMAST for &period
            INDEX ON SN TO XTMAST
            USE XTMAST INDEX XTMAST
            Set relation to sn into B
            Copy to STMAST for b->class='&classn'
            Use STMAST
            Erase xtmast.dbf
            Erase xtmast.ndx
            Exit
        Case upper(chr(i))="3"
            Sel="3"
            @ 16,30 Say "Enter stock number" get stockn;
            pict '9999-99-999-9999'
            Read
            Select A
            Use MlMAST
            Index on sn to MlMAST
            Use MlMAST INDEX MlMAST
            Store "&period .AND. sn='&stockn'" to condit1
            Copy to STMAST for &condit1
            Exit
        Case upper(chr(i))="0"
            Close databases
            Return
        Otherwise
            ? chr(7)
    Endcase
Enddo
Select A
Use STMAST
Index on SN+TYPE+CI to STMAST
Use STMAST index stmast
Erase MlMAST.dbf
Erase MlMAST.ndx
Do printer

```



```

Store 1 to linectr,pagectr,N,liner
Store 0 to mdi,reqqty,canqty,recqty,turnqty,DInow,mqty
Store 0 to sreqqty,smdi,scanqty,srecqty,sturnqty,sdinow
If Eof()
  @ 10,18 Clear to 14,58
  @ 10,18 to 14,58
  @ 11,20 SAY "    You Never Requested"
  @ 13,20 Say "    Press any key to return...."
  Wait ""
  Close databases
  Erase SIMAST.DBF
  Erase SIMAST.NDX
  Return
Endif
select A
Clear
Go top
*.....Holds one of item from property file
Do while .NOT. Eof()
  Store SN to stockn
  select B
  store class to classn
  Store nm to mnm
  Select A
  Seek stockn
  If found()
    Sum qty*Unitcost for type="RL" .AND. date < begdate
    Store qty*unitcost to RLqty
    Sum Qty*unitcost for type="IL" .AND. date < begdate
    Store qty to ILqty
    MDI=RLqty - ILqty
    Seek stockn
    Do while .NOT. Eof() .AND. sn=stockn
      Store qty*unitcost to mqty
      Do case
        Case type = "RL"
          Reqqty = reqqty+mqty
        Case Type = "CL"
          Canqty = Canqty+mqty
        Case Type = "IL"
          Recqty = Recqty+mqty
        Case Type = "TL"
          Turnqty = Turnqty+mqty
      Endcase
      Skip
    Enddo
    Store MDI+reqqty-Canqty-Recqty to DInow
    If liner=1
      ? |      |
      ? |      |
      ? |      |
      ? |      |
      ?? |      |      Transact Status Report'
      ? |      |      Page ' + str(pagectr,2)
      ?? |      |      Date : '+Dtoc(date())
      ? | From : The 150 Infantry Division'
      ? | To : The 3333 Logistics Supprot Command'
      ? |      |
      ?? | From      To      |
      ? |      |      |
      ?? | Dtoc(begdate)+' - '+Dtoc(enddate)
      ? |      |      |
      ?? | 'No      Stock number  Description '
      ?? |      D/Ibeg  Req st Cancel Rec ve Turnin D/Inow'
      Store pagectr + 1 to pagectr
    Endif
    Store linectr to Z
    ? Str(Z,4)+' '+stockn+' '+rtrim(mnm)+space(15 - LEN(rtrim(mnm)))
  Endwhile

```



```

?? '+'+substr(str(MDI),5,6)+' '+'+ substr(Str(reqqty),4,7)
?? substr(str(Canqty),4,7)+substr(str(Recqty),4,7)
?? substr(str(turnqty),4,7)+substr(Str(DInow),4,7)
Store Smdi+mdi to Smdi
Store Sreqqty+reqqty to Sreqqty
Store Scanqty+canqty to Scanqty
Store Srecqty+recqty to Srecqty
Store Sturnqty+turnqty to Sturnqty
Store Sdinow+dinow to Sdinow
Store 0 to mdi, reqqty, canqty, recqty, turnqty, DInow
Linectr=linectr+1
Liner= liner+1
If liner=40*N
    ? '
    ?? '
    ? '
    ?? '          Grand Total '+' str(total)
    ? '
    ? '
    ? '
    ? 'Material Management NCO : _____'
    ?? 'Date : ____/____/____'
    ? 'Material Management Officer : _____'
    ?? 'Date : ____/____/____'
    Eject
    N = N + 1
Endif
Endif
If sel="1"
    If linectr=1
        Select B
        Skip
        Store class to classn
        Loop
    Endif
    Select B
    Skip
Endif
If B->Class<>Classn .OR. Eof()
    ? 'Class '+'classn+'
    ?? '-----'
    ? '
    ?? 'Subtotal '+' '+'+substr(str(smdi),5,6)+' '+'
    ?? substr(str(sreqqty),4,7)+ substr(str(Scanqty),4,7)
    ?? Substr(str(Srecqty),4,7)+Substr(Str(Sturnqty),4,7)
    ?? substr(Str(Sdinow),4,7)
    ? '
    Store 0 to sreqqty, smdi, scanqty, srecqty, sturnqty, sdinow
    Store 1 to linectr
    Liner=liner+3
Endif
Enddo
If liner>1
    ? '
    ? '
    ? 'Matrrial Management NCO : _____'
    ?? 'Date : ____/____/____'
    ? '
    ? 'Material Management Officer : _____'
    ?? 'Date : ____/____/____'
    linectr = 1
    pagectr = 1
Endif
If mprint="Y"
    Eject
    Set console on
    Set print off
Endif
Set color to N/W
Wait

```

```

Set color to
Store space(16) to stockn
Clear
Close Databases
Erase STmast.dbf
Erase stmast.ndx
If sel="1"
    Erase csproper.ndx
Endif
@ 9,15 to 15,64 double
@ 11,25 Say "Finished report for TSR "
@ 13,25 Say "Period "+Dtoc(begdate)+" - "+Dtoc(enddate)
N=0
Do while n<35
    N=N+1
Enddo
Clear
Store Space(16) to stockn
Release all
Return
*-----* Eof RPTSR.prg *-----*

```

c. RPOST

```

*****
***** RPOST.PRG *****
*****
* Module name.....: RPOST.prg *
* Author.....: Park, Taeyong *
* Date.....: Sept 25. 1987 *
* Purpose.....: Generate Oder shipping time report *
* Called by.....: Reports.prg *
* Modules called : None *
*****
Clear
Select B
Use PROPERTY index snproper
Do while .T.
    Store " " to Classn, Mprint
    Store "ORDER SHIPPING TIME REPORT" to mtitle
    Do select
        Store "(Date>=Begdate .AND. Date<=Enddate)" to period
    Do case
        Case upper(Chr(i))="1"
            Sel="1"
            Select B
            Index on Class+Sn to CSPPROPER
            Select A
            Use master
            Store "type='RL' .AND.&period" to condit1
            Copy to RLMAST for &condit1
            Store "type='IL' .AND.&period" to condit2
            Copy to ILMASST for &condit2
            Exit
        Case upper(chr(i))="2"
            Sel="2"
            @ 14,35 Say "Enter Class number(1-10) " get Classn
            @ 15,35 Say " (0 for 10)"
            Read
            If classn = "0"
                Store "10" to Classn
            Endif
            Select A
            Use master INDEX sctmast
            Set relation to sn into B
            Store "type='RL' .AND. &period .AND. b->class='&classn'";

```

```

        to condit1
        Copy to RLMAST for &condit1
        Store "type='IL' .AND. &period .AND. b->class='&classn'";
        to condit2
        Copy to ILMAST for &condit2
        Exit
    Case upper(chr(i))="3"
        Sel="3"
        @ 16,35 Say "Enter stock number" get stockn;
        pict '9999-99-999-9999'
        Read
        Select A
        Use master INDEX sctmast
        Set relation to sn into B
        Store "type='RL' .AND. &period .AND. sn='&stockn'";
        to condit1
        Copy to RLMAST for &condit1
        Store "type='IL' .AND. &period .AND. sn='&stockn'";
        to condit2
        Copy to ILMAST for &condit2
        Exit
    Case upper(chr(i))="0"
        Close databases
        Return
    Otherwise
        ? chr(7)
Endcase
Enddo
Select A
Use RLMAST
Index on SN+VN to SRRL
Use RLMAST index SRRL
If Eof()
    @ 10,18 Clear to 14,58
    @ 10,18 to 14,58
    @ 11,20 SAY "    You Never Requested"
    @ 13,20 Say "    Press any key to return...."
    Wait ""
    Close databases
    Erase RLMAST.DBF
    Erase SRRL.NDX
    Return
Endif
Select C
Use ILMAST
Index on SN+VN to SRIL
Use ILMAST index SRIL
If Eof()
    @ 9,18 clear to 13,60
    @ 9,18 to 13,60
    @ 10,20 SAY "You Never Received the requested item"
    @ 12,20 Say "Press any key to return...."
    Wait ""
    Close databases
    Erase RLMAST.DBF
    Erase SRRL.NDX
    Erase ILMAST.DBF
    Erase SRIL.NDX
    Return
Endif
Store 0 to SN1,SN2,SN3,SN4,SN5,SM1QTY,SM2QTY,SM3QTY,SM4QTY,SM5QTY
Store 0 to N1,N2,N3,N4,N5,M1QTY,M2QTY,M3QTY,M4QTY,M5QTY
Store 0 to STOIN,TOIQTY,TOIN,STOTQTY,mdate
Store 1 to linectr,pagectr,N,liner
Go top
Do printer
Clear
select A
Go top

```

```

*.....Holds one of item from property file
Do while .NOT. Eof()
  Store SN to stockn
  select B
  store class to classn
  Select C
  *.....Find the stock number which is the same as
  *.....at the file which is sorted type "IL"
  Seek stockn
  If found()
    Do while .NOT. Eof() .AND. sn=stockn
      Store Date to mldate
      Store VN to M1VN
      Store Stockn+M1VN to mstock
      *.....Find the stocknumber which is the same voucher
      *.....at the file which is sorted by type "RL"
      Select A
      Seek mstock
      If found()
        Store date to m2date
        Store qty to mqty
        Store mldate-m2date to mdate
        Do case
          Case mdate<7
            N1=N1+1
            M1qty=M1qty+mqty
          Case mdate-7<7
            N2=N2+1
            M2qty=M2qty+mqty
          Case (Mdate>=15 .and. Mdate<30)
            N3=N3+1
            M3qty=M3qty+mqty
          Case (Mdate>=30 .and. Mdate<60)
            N4=N4+1
            M4qty=M4qty+mqty
          Case (Mdate>=60 .and. Mdate<90)
            N5=N5+1
            M5qty=M5qty+mqty
        Endcase
      Endif
      Store N1+N2+N3+N4+N5 to TOTN
      Store M1QTY+M2QTY+M3QTY+M4QTY+M5QTY to TOTQTY
      Select C
      Skip
    Enddo
  If liner=1
    ? |
    ? |
    ? |
    ? |
    ?? | Report
    ? |
    ?? |
    ? |
    ? | To : The 3333 Logistics Support Command'
    ? | From : The 150 Infantry Division'
    ? |
    ? |
    ?? | To '
    ? |
    ?? | Dtoc(enddate)+' )'
    ? |
    ? | No Stock number '
    ?? | Total 7days 14days 30days '
    ?? | 60days'
    ? |
    ?? | Total Q ty Freq Q ty Freq Q ty Freq Q ty '
    ?? | Freq Q ty'
    store pagectr + 1 to pagectr
  
```

```

Endif
Store linectr to Z
? Str(Z,3)+' '+stockn+substr(str(TOTN),5,6)
?? substr(str(TOTQTY),5,6)
?? substr(str(N1),5,6)+substr(Str(M1QTY),5,6)
?? substr(str(N2),5,6)+substr(str(M2QTY),5,6)
?? Substr(str(N3),5,6)+substr(str(M3QTY),5,6)
?? substr(str(N4),5,6)+substr(Str(M4QTY),5,6)
Store STOTN+TOTN to STOTN
Store STOTQTY+TOTQTY to STOTQTY
Store SN1+N1 to SN1
Store SM1QTY+M1QTY to SM1QTY
Store SN2+N2 to SN2
Store SM2QTY+M2QTY to SM2QTY
Store SN3+N3 to SN3
Store SM2QTY+M2QTY to SM2QTY
Store SN4+N4 to SN4
Store SM4QTY+M4QTY to SM4QTY
Linectr=linectr+1
liner= liner+1
If liner=40*N
? '
? '
? '
? '
? 'Material Management NCO : _____'
?? 'Date : ____/____/____'
? 'Material Management Officer : _____'
?? 'Date : ____/____/____'
Eject
N = N + 1
Endif
Endif
If sel="1"
If linectr=1
Select B
Skip
Store class to classn
Loop
Endif
Select B
Skip
Endif
If (b->class<>classn) .OR. EOF()
? 'Class '+classn+ '-----'
?? '-----'
? 'Subtotal '
?? substr(str(STOTN),5,6)+substr(str(STOTQTY),5,6)
?? substr(str(SN1),5,6)+ substr(Str(SM1QTY),5,6)
?? substr(str(SN2),5,6)+substr(str(SM2QTY),5,6)
?? Substr(str(SN3),5,6)+substr(str(SM3QTY),5,6)
?? substr(str(SN4),5,6)+substr(Str(SM4QTY),5,6)
? '
Store 0 to SN1,SN2,SN3,SN4,SN5,SM1QTY,SM2QTY,SM3QTY,SM4QTY,SM5QTY
Store 0 to N1,N2,N3,N4,N5,M1QTY,M2QTY,M3QTY,M4QTY,M5QTY
Store 0 to STOTN,TOTQTY,TOTN,STOTQTY,mdate
Store 1 to linectr
Liner=liner+3
Endif
Enddo
If liner > 1
? '
? '
? 'Matrial Management NCO : _____'
?? 'Date : ____/____/____'
? '
? 'Material Management Officer : _____'
?? 'Date : ____/____/____'
Endif

```



```

If mprint="Y"
  Eject
  Set console on
  Set print off
Endif
Close databases
Erase Csproper.ndx
Erase RLMAST.DBF
Erase SRRL.NDX
Erase ILMASST.DBF
Erase SRIL.NDX
Set color to N/W
Wait
Set color to
@ 9,15 clear to 15,64
@ 9,15 to 15,64 double
@ 11,25 Say "Finished report for OST "
@ 13,25 Say "Period "+Dtoc(begdate)+" - "+Dtoc(enddate)
N=0
Do while n<35
  N=N+1
Enddo
Store Space(16) to stockn
Release all
Clear
Return
*-----* Eof OST.prg *-----*

```

d. RPEIS

```

*****
*****                                RPEIS.PRG                                *****
*****
* Module name.....: RPEIS.prg                                           *
* Author.....: Park, Taeyong                                           *
* Date.....: Sept 24, 1987                                             *
* Purpose.....: Generate Combat essential item                         *
*               Stockout history report to higher command.             *
* Called by.....: Reports.prg                                           *
* Modules called:                                                       *
* Variables used.:                                                       *
*               Public.:                                                *
*               Local...: S, Sts, Mprint, Mqty, Mcost,                  *
*               *                                                       *
*****
Select B
Use PROPERTY
Copy to TEMPPROP for ESSENCE
Use TEMPPROP
Index on Class+SN to CSproper
Use TEMPPROP index CSPROPER
Select A
Use STOCKOUT
Index on SN to SNSTOK
Use STOCKOUT index SNSTOK
Select B
Store 1 to linectr,pagectr,N,liner
Store 0 to sts30,sts60,sts90,sts120,stco30,stco60,stco90,stcol20
Store 0 to s30,s60,s90,s120,sco30,sco60,sco90,scol20,msqty,mscost
Go top
Store CLASS to CLASSN
Store " " to Mprint
Do printer
Clear
Do while .NOT. Eof()
  Store SN to STOCKN
  Store NM to MNM

```



```

Select A
Seek STOCKN
If found().
  Do while .NOT. Eof() .AND. sn=stockn
    Store QTY to MQTY
    Store QTY*UNITCOST to MCOST
    If Dtoc(REFILLDATE)=" "
      Skip
    Loop
  Endif
  Store REFILLDATE to FILLDATE
  Store OUTDATE to SOUTDATE
  Bal = FILLDATE-SOUTDATE
  Do case
    Case bal <= 30
      S30 = S30+mgty
      Sco30 = Sco30+mcost
    Case Bal <= 60 .AND. Bal>30
      S60 = S60+mgty
      Sco60 = Sco60+mcost
    Case Bal <= 90 .AND. Bal>60
      S90 = S90+mgty
      Sco90 = Sco90+mcost
    Case Bal > 90
      S120 = S120+mgty
      Sco120 = Sco120+mcost
  Endcase
  Skip
Enddo
Store S30+S60+S90+S120 to SQTY
Store SCO30+SCO60+SCO90+SCO120 TO SCOST
If liner=1
  ? |
  ? |
  ? |
  ? |
  ?? 'Stockout Report'          Combat Rssential Item '
  ? |                               Page ' + str(pagectr,2)
  ?? '
  ? |                               Date : '+Dtoc(date())
  ? |
  ? |   To : The 3333 Logistics Supprot Command'
  ? |   From : The 150 Infantry Division'
  ? |
  ? |                               From           To '
  ? |                               ('+Dtoc(begdate)+' ) - ('
  ?? Dtoc(enddate)+' )'
  ? |
  ? |   No      Stock number      '
  ?? '   Total      30days      60days      90days      '
  ?? '120days'
  ? |
  ?? 'Total Cost  Item  Cost  Item  Cost  Item  Cost  '
  ?? 'Item  Cost'
  ? |
  store pagectr + 1 to pagectr
Endif
Store linerctr to Z
? Str(Z,3)+' '+stockn+substr(str(sqty),5,6)
?? substr(str(scost),5,6)
?? substr(str(S30),5,6)+substr(str(sco30),5,6)
?? substr(str(s60),5,6)+substr(str(sco60),5,6)
?? substr(str(s90),5,6)+substr(str(sco90),5,6)
?? substr(str(s120),5,6)+substr(str(sco120),5,6)
Store msqty+sqty to msqty
Store mscost+scost to mscost
Store Sts30+s30 to Sts30
Store Stco30+sco30 to Stco30

```

```

Store Sts60+s60 to Sts60
Store Stco60+sco60 to Stco60
Store Sts90+s90 to Sts90
Store Stco90+sco90 to Stco90
Store Sts120+s120 to Sts120
Store Stcol20+scol20 to Stcol20
Linectr=linectr+1
liner= liner+1
If liner=40*N
? '
? '
? '
? '
? 'Material Management NCO : _____'
?? 'Date : ____/____/____'
? 'Material Management Officer : _____'
?? 'Date : ____/____/____'
Eject
N = N + 1
Endif
Endif
If linectr=1
Select B
Skip
Store Class to classn
Loop
Endif
Select B
skip
If class<>classn .OR. Eof()
? 'Class '+classn+' -----'
?? '-----'
? 'Subtotal '
?? substr(str(msqty),5,6)+substr(str(mscost),5,6)
?? substr(str(Sts30),5,6)+ substr(Str(stco30),5,6)
?? substr(str(sts60),5,6)+substr(str(stco60),5,6)
?? Substr(str(sts90),5,6)+substr(str(stco90),5,6)
?? substr(str(sts120),5,6)+substr(Str(stcol20),5,6)
? '
? '
Store 0 to sts30,sts60,sts90,sts120,stco30,stco60,stco90,stcol20
Store 0 to S30,s60,s90,s120,sco30,sco60,sco90,scol20,mscost,msqty
Store 1 to linectr
Liner=liner+4
Endif
Store class to classn
Enddo
If liner > 1
? '
? '
? 'Matrial Management NCO : _____'
?? 'Date : ____/____/____'
? '
? 'Material Management Officer : _____'
?? 'Date : ____/____/____'
Endif
If mprint ="Y"
Eject
Set print off
Set console on
Endif
Close all
Erase SNstok.ndx
Erase tempprop.dbf
Erase csproper.ndx
@ 9,15 clear to 15,64
@ 9,15 to 15,64
@ 11,25 Say "Finished report for Stockout "
@ 13,25 Say "Period "+Dtoc(begdate)+" - "+Dtoc(enddate)
N=0

```

Do while n<35

N=N+1

Enddo

Release all

Clear

Return

----- Eof Eis.prg *-----*

e. RPSOR

```
*****
***** RPSOR.PRG *****
*****
* Module name.....: RPSOR.prg *
* Author.....: Park, Taeyong *
* Date.....: Sept 24. 1987 *
* Purpose.....: Generate Stockout history report *
*               to higher command. *
* Called by.....: Reports.prg *
*****
```

Select B

Use PROPERTY index snproper

clear

Do while .T.

Store " " to Classn, Mprint

Store " STOCKOUT REPORT" to mtitle

Do select

Store "(OutDate>=Begdate .AND. outDate<=Enddate)" to period

Do case

Case upper(chr(i))="1"

Sel="1"

Select B

Index on Class+Sn to CSPROPER

Select A

Use stockout

Copy to TEMPOUT for &period

Exit

Case upper(chr(i))="2"

Sel="2"

@ 14,30 Say "Enter Class number(1-10) " get Classn

@ 15,30 Say " (0 for 10)"

Read

If classn = "0"

Store "10" to Classn

Endif

Select A

Use stockout INDEX stockout

Set relation to sn into B

Store "&period .AND. b->class='&classn'" to condit1

Copy to TEMPOUT for &condit1

Exit

Case upper(chr(i))="3"

Sel="3"

@ 16,30 Say "Enter stock number" get stockn;

pict '9999-99-999-9999'

Read

Select A

Use stockout INDEX stockout

Set relation to sn into B

Store "&period .AND. sn='&stockn'" to condit1

Copy to TEMPOUT for &condit1

Exit

Case upper(chr(i))="0"

Close databases

Return

Otherwise

? chr(7)

```

Endcase
Enddo
Select A
Use TEMPOUT
Index on SN to SNTEMP
Use TEMPOUT index SNTEMP
If Eof()
    @ 10,18 Clear to 15,58
    @ 10,18 to 15,58
    @ 11,20 SAY "      Never Stockout"
    @ 13,20 Say "      Press any key to return...."
    Wait ""
    Close databases
    Erase TEMPOUT.DBF
    Erase SNTEMP.NDX
    Return
Endif
Do printer
Clear
Store 1 to linectr,pagectr,N,liner
Store 0 to sts30,sts60,sts90,sts120,stco30,stco60,stco90,stco120
Store 0 to S30,s60,s90,s120,sco30,sco60,sco90,sco120,msqty,mscost
Go top
Do while .NOT. Eof()
    Store sn to stockn
    Select B
    Store nm to mnm
    Store class to classn
    Select A
    Seek stockn
    If found()
        Do while .NOT. Eof() .AND. SN=STOCKN
            Store qty to mgty
            Store qty*unitcost to mcost
            If Dtoc(refilldate)=" "
                Skip
            Loop
        Endif
        Store refilldate to filldate
        Store outdate to soutdate
        Bal = (filldate)-(soutdate)
        Do case
            Case bal <= 30
                S30 = S30+mgty
                Sco30 = Sco30+mcost
            Case Bal <= 60 .AND. Bal>30
                S60 = S60+mgty
                Sco60 = Sco60+mcost
            Case Bal <= 90 .AND. Bal>60
                S90 = S90+mgty
                Sco90 = Sco90+mcost
            Case Bal > 90
                S120 = S120+mgty
                Sco120 = Sco120+mcost
        Endcase
        Skip
    Enddo
    Store S30+S60+S90+S120 to SQTY
    Store SCO30+SCO60+SCO90+SCO120 TO SCOST
    If liner=1
        ? |
        ? |
        ? |
        ? |
        ?? |
        ? |
        ?? |
        ? |
    Endif

```

```

                                Stockout Report'
Page ' + str(pagectr,2)
                                Date : '+Dtoc(date())

```

```

? ' To : The 3333 Logistics Supprot Command'
? ' From : The 150 Infantry Division'
? '
? '
? ' No Stock number '
?? ' Total 30days 60days 90days '
?? '120days'
? '
?? 'Total Cost Item Cost Item Cost Item Cost '
?? 'Item Cost'
store pagectr + 1 to pagectr
Endif
Store linectr to Z
? Str(Z,3)+' '+stockn+substr(str(sqty),5,6)
?? substr(str(scost),5,6)
?? substr(str(S30),5,6)+substr(Str(sco30),5,6)
?? substr(str(s60),5,6)+substr(str(sco60),5,6)
?? Substr(str(s90),5,6)+substr(str(sco90),5,6)
?? substr(str(s120),5,6)+substr(Str(scol20),5,6)
Store msqty+sqty to msqty
Store mscost+scost to mscost
Store Sts30+s30 to Sts30
Store Stco30+sco30 to Stco30
Store Sts60+s60 to Sts60
Store Stco60+sco60 to Stco60
Store Sts90+s90 to Sts90
Store Stco90+sco90 to Stco90
Store Sts120+s120 to Sts120
Store Stcol20+scol20 to Stcol20
Linectr=linectr+1
liner= liner+1
If liner=40*N
? '
? '
? '
? '
? 'Material Management NCO : _____'
?? 'Date : ____/____/____'
? 'Material Management Officer : _____'
?? 'Date : ____/____/____'
Eject
N = N + 1
Endif
Endif
If sel="1"
If linectr=1
Select B
Skip
Store Class to classn
Loop
Endif
Select B
skip
Endif
If b->class<>classn .OR. Eof()
? 'Class '+classn+' -----'
?? '-----'
? ' Subtotal '
?? substr(str(msqty),5,6)+substr(str(mscost),5,6)
?? substr(str(Sts30),5,6)+ substr(Str(stco30),5,6)
?? substr(str(sts60),5,6)+substr(str(stco60),5,6)
?? Substr(str(sts90),5,6)+substr(str(stco90),5,6)
?? substr(str(sts120),5,6)+substr(Str(stcol20),5,6)
? '
Store 0 to sts30,sts60,sts90,sts120,stco30,stco60,stco90,stcol20
Store 0 to S30,s60,s90,s120,sco30,sco60,sco90,scol20,mscost,msqty
Store 1 to linectr
Liner=liner+3
Endif

```

```

Enddo
If liner > 1
    ? |
    ? |
    ? 'Matrrial Management NCO : _____'
    ?? 'Date : ____/____/____'
    ? |
    ? 'Material Management Officer : _____'
    ?? 'Date : ____/____/____'
Endif
Close all
If mprint = "Y"
    Eject
    Set console on
    Set print off
Endif
If sel="1"
    Erase csproper.ndx
Endif
Erase tempout.dbf
Erase SNTemp.ndx
Set color to N/W
Wait
Set color to
@ 9,15 clear to 15,64
@ 9,15 to 15,64 double
@ 11,25 Say "Finished report for Stockout "
@ 13,25 Say "Period "+Dtoc(begdate)+" - "+Dtoc(enddate)
N=0
Do while n<35
    N=N+1
Enddo
Clear
Store space(16) to stockn
Return
*-----* Eof SOR.prg *-----*

```


4. ANALYSIS

a. ANALYSIS STARTUP

```
*****
***** ANALYSIS.PRG *****
*****
* Module name.....: ANALYSIS.prg
* Author.....: Park, Taeyong
* Date.....: Sept 25. 1987
* Purpose.....: Analyze the reorder point, safety level
*               EOQ by using probabilistic model
*               (Fixed Order Size System)
* Called by.....: PMAIN.prg
* Modules called : ANEQQ.prg, ANLEADT.prg
* Variables used.:
*   Public..: TODAY.STD.STOCKN
*   Local..: BEGDATE, ENDDATE, CTITLE, MLEADT, MQTY, ORDERC,
*            SERVL, FRATE, RSTD, LSTD, N, M, CSC, CSL.
*****
```

```
DO WHILE .T.
  Store space(8) to BEGDATE, ENDDATE
  Store space(50) to CTITLE
  Store "N" to CSL, CSC
  Store space(16) to STOCKN, MREQNO
  Store 0 to MLEADT, MQTY, RDEM, ORDERC, SCOST, SERVL, FRATE, RSTD, LSTD
  Store 0 to DATEBAL, LEADDEM, TOTDATE, MLEADT1
  Store 1 to N, M
  CLEAR
  Text
```

THIS SYSTEM USES THE FIXED ORDER SIZE SYSTEM WITH PROBABILISTIC MODEL. THE FIXED ORDER SIZE SYSTEM IS COMPLETELY DEFINED BY THE ORDER QUANTITY $Q|$ AND REORDER POINT $B|$. THE RISK OF STOCKOUT OCCURS AFTER REORDER POINT. TO GET $Q|, B|$, YOU HAVE TO DECIDE SERVICE LEVEL, OR STOCKOUT COST PER UNIT. THIS SYSTEM ASSUMED BACKORDER CASE WITH SERVICE LEVEL INSTEAD OF LOSTSALES CASE.

```
Endtext
@ 19,17 SAY "Enter stock number : "
@ 21,12 SAY "Enter time period you want to;
test(
SET COLOR TO N/W
@ 21,48 SAY "01/01/87"
@ 21,59 SAY "12/31/87"
SET COLOR TO
@ 22,50 SAY "Begin      End date"
@ 24,20 SAY "Leave BLANK any space to Exit "
@ 1,10 TO 3,60 DOUBLE
@ 4,1 TO 23,77 DOUBLE
@ 2,15 SAY "A N A L Y S I S  of T R A N S A C T I O N"
@ 3,62 SAY "Today : " +DTOC(DATE())
@ 19,38 GET STOCKN PICT "9999-99-999-9999"
READ
IF STOCKN=' '
  EXIT
ENDIF
@ 21,48 GET BEGDATE PICT "99/99/99"
@ 21,59 GET ENDDATE PICT "99/99/99"
Read
If Begdate=" " .OR. Enddate=" " .OR. Stockn=" "
  Exit
Endif
```

```

Store Ctod(Begdate) to Begdate
Store Ctod(Enddate) to Enddate
If BEGDATE>=ENDDATE
    Clear
    @ 15, 25 SAY "The End date MUST be later than the begin date"
    @ 17, 30 SAY "Press ENTER to redo....."
    Wait " "
    Loop
Endif
Do ANLEADT
If Rdem<=0
    Clear
    Loop
Endif
Do ANEOQ
ENDDO
CLOSE DATABASES
RELEASE ALL
STOCKN=SPACE(16)
RETURN
*-----* Eof: Analysis.prg *-----*

```

b. ANLEADT

```

*****
***** LEADT.PRG *****
*****
* Module name.....: Leadt.Prg *
* Author.....: Park, Taeyong *
* Date.....: Sept 30. 1987 *
* Purpose.....: Calculate Lead time to Supply for given Item *
* Called by.....: Analysis.Prg *
* Modules called : Proc.prg *
* Variables used..: *
*      Public.: MleadT,Mqty,Mzulu *
*      Local.: *
*****
Set procedure to ANPROC
SET CONSOLE OFF
Clear
@ 10,15 to 14,65
@ 11,25 Say " Please do NOT touch"
@ 13,25 Say " Working..... "
Select B
Use PROPERTY index snproper
Select A
Use master index sctmast
Copy to RLMAST for type="RL" .AND. (Date>=Begdate .AND. Date<=Enddate);
      .AND. sn=stockn
Copy to ILMASST for type="IL" .AND. (Date>=Begdate .AND. Date<=Enddate);
      .AND. sn=stockn
Copy structure to DEM fields qty,type,date
Select E
Use DEM
Select C
Use ILMASST
Index on REQNO to SRIL
Use ILMASST index SRIL
Select D
Use RLMAST
Index on REQNO to SRRL
Use RLMAST index SRRL
Select C
If Eof()
    ? chr(7)

```

```

@ 15, 25 SAY "You receive nothing from LSC between"
@ 16, 30 SAY Dtoc(Begdate)+" - " +Dtoc(Enddate)
@ 18, 25 SAY "Press ENTER to return...."
Wait " "
Close databases
Erase RLMAST.DBF
Erase SRRL.NDX
Erase ILMASST.DBF
Erase SRIL.NDX
Erase DEM.DBF
Release all
Set console on
Return
Endif
*.....To calculate the lead time of an item
Go top
Do while .NOT. Eof()
  Store regno to MREQNO
  Store date to ILDATE
  Select D
  Seek Mreqno
  If Eof() .OR. Bof()
    Select C
    Skip
    Loop
  Else
    Store Date to RLDATE
    Do while regno=mreqno .AND. .NOT. Eof()
      N=N+1
      Datebal=(c->Date) - (Date)
      Totdate=Totdate+Datebal
      If rldate>date
        Store Date to Rldate
      Endif
      Skip
    Enddo
  Endif
  Select A
  Sum qty for sn=stockn .AND. type="RD" .AND.;
  (date>=RLDATE .AND. date<=ILDATE) to leaddem
  Select E
  Append blank
  Replace qty with leaddem
  Replace type with "LD"
  Replace date with Rldate
  Store 0 to leaddem
  Select C
  Skip
Enddo
Store Totdate/N to Mleadtl
store Mleadtl/30 to mleadtl
Store 0 to N
*.....To get Annual average demand (1 year moving average)
Select A
Store "date>=begdate .AND. date<=enddate" to Condx
Sum qty for sn=Stockn.AND.type="RD".AND. &condx to Rdem
Store enddate-begdate to totd
Store Rdem*365/totd to Rdem
Store Round(totd/30,0) to Nmax
*.....To get standard deviation of annual demand from monthly demand
Go top
Store begdate to mdate
Store "(sn=stockn.and.type='RD') " to cond1
Store "(date>=mdate+30*N .AND. date<=mdate+30*(N+1))" to cond2
Do while N < Nmax
  Sum qty for &cond1 .AND. &cond2 to mqty
  Select E
  Append blank

```

```

Replace qty with mqty
Replace date with (mdate+30*n)
Replace type with "DM"
N= N+1
Select A
Store 0 to mqty
Enddo
Select E
Do Std with "qty","Type='DM'"
Close databases
Erase RLMAS.T.DBF
Erase SRRL.NDX
Erase ILMAS.T.DBF
Erase SRIL.NDX
Erase DEM.DBF
SET CONSOLE ON
SET PROCEDURE TO
RETURN
*-----* Eof ANLEADT.prg *-----*

```

c. ANEOQ

```

*****
***** ANEOQ.PRG *****
*****
* Module name....: ANEOQ.prg *
* Author.....: Park, Taeyong *
* Date.....: OCT 1. 1987 *
* Purpose.....: To calculate Economic Order Quantity, Reorder *
*               Point, Safety level, and Requisition Object *
* Called by.....: ANALYSIS.prg *
* Modules called : None *
* Variables used..: *
*               Public.: STD, TODAY, *
*               Local..: CSC.CSL.RDEM.MEOFZ.MTZ.MSERVL.ORDERC.FRAT *
* *
*****
Set decimals to 5
Select C
Use ASL index INASL

Select B
Use NORMAL
Select A
Use PROPERTY index snproper
Set relation to SN into C
Do while .T.
    Seek stockn
    Store "PRESENT" to title
    Clear
    Store "Is this record right?(Y/N):" to ctitle
    Store Ltrim(str(ASL->ROP)) to Mrop
    Store Ltrim(str(ASL->SL)) to Msl
    Store Ltrim(str(ASL->RO)) to Mro
    Store Ltrim(str(ASL->OST)) to Most
    Do while .T.
        Set format to EQQ
        Read
        If check = "Y"
            Set format to
            Exit
        Endif
    Enddo
    *.....Holding cost
    If Frate=0
        Store 15.0 to Frate
    Endif

```

```

Store (Frate/100)*Property->unitcost to Holdc
*.....ordering cost
If Orderc = 0
    Store 20.0 to Orderc
Endif
Do case
    Case (CSL="Y" .AND. CSC="Y").OR.(CSL="N" .AND. CSC="N")
        ? Chr(7)
        @ 10, 15 clear to 16,65
        @ 10,15 to 14,65
        @ 12,20 SAY "You have to select ONE of Choices"
        @ 15,20 Say "Press ENTER to redo....."
        Store 0 to Frate,Orderc,Scost,Servl
        Store "N" to CSC,CSN
        Store "X" to check
        Wait " "
        Loop
    *.....Holding cost rate
    Case CSL="N" .AND. CSC="Y"
        Q0 = Sqrt(2*Rdem*Orderc/Holdc)
        Msigfofb=Holdc*Q0/(scost*Rdem)
        Select B
        Locate for sigfofb <=msigfofb
        Store Tz to Mtz
        Store Eofz to meofz
    Case CSL="Y" .AND. CSC="N"
        If Servl=0
            Store 85.000 to servl
        Endif
        Store servl/100 to mServl
        Q0 = Sqrt(2*Rdem*Orderc/Holdc)
        Scost=Holdc*Q0/(mServl*Rdem)
        Store 1-Servl to mpofs
        Select B
        Locate For pofs<=mpofs
        Store Tz to Mtz
        Store Eofz to meofz
Endcase
Store 0 to EQQ,MOST,MROP,MSL,MRO
Store Sqrt(2*Rdem*(Orderc+Scost*Meofz))/Holdc to EQQ
Store Mleadtl to Most
Store Rdem*Mleadtl/12 + Mtz*Std*Sqrt(Mleadtl) to Mrop
Store Mrop - (Rdem*Mleadtl/12) to Msl
Store EQQ+Mrop to Mro
Store "Calculated" to title
Store "Do you want to change the old records with this?:" to ctitle
Select C
Store Ltrim(str(MROP)) to Mrop
Store Ltrim(str(MSL)) to Msl
Store Ltrim(str(MRO)) to Mro
Store Ltrim(str(MOST)) to Most
Seek stockn
Set format to EQQ
Read
If check ="Y"
    Set format to
    Replace RO with Val(Mro)
    Replace Rop with Val(Mrop)
    Replace SL with Val(Msl)
    Replace OST with Val(Most)
    Replace date with today
    Exit
Else
    Set format to
    Store 0 to Frate,Orderc,Scost,Servl
    Store "N" to CSC,CSN
    Store "X" to check
Endif

```



```

Enddo
Close databases
Set decimals to
Return
*-----* Eof EOQ.prg *-----*

```

d. ANPROC

```

*****
***** PROC.PRG *****
*****
* Module name.....: Proc.prg *
* Author.....: Park, Taeyong *
* Date.....: Sept 30. 1987 *
* Purpose.....: Procedures for calaulating statics *
* Variables used.: *
*      Public.: FieldName *
*      Local.: Max,Min,Std,Var *
*****

PROCEDURE Max
PARAMETERS fieldname,Condition
  SET FILTER TO &Condition
  GO TOP
  Max = -99999
  DO While .NOT. EOF()
    IF &FieldName > Max
      Max = &FieldName
    Endif
    Skip
  ENDDO
  SET FILTER TO
RETURN

PROCEDURE Min
PARAMETERS FieldName,Condition
  SET FILTER TO &Condition
  GO TOP
  Min = 99999
  DO While .NOT. EOF()
    IF &FieldName < Min
      Min = &FieldName
    Endif
    Skip
  ENDDO
  SET FILTER TO
RETURN

PROCEDURE Var
PARAMETERS FieldName,Condition
  SET FILTER TO &Condition
  GO TOP
  COUNT TO N
  SUM(&FieldName) TO TOT
  SUM(&FieldName 2) TO TOTsq
  Correction = TOT 2/N
  Var = (TOTsq - Correction)/(n-1)
  SET FILTER TO
RETURN

PROCEDURE Std
PARAMETERS FieldName,Condition
  SET FILTER TO &Condition
  GO TOP
  COUNT TO N
  SUM(&FieldName) TO TOT
  SUM(&FieldName 2) TO TOTsq
  Correction = TOT 2/N
  Variance = (TOTsq - Correction)/(n-1)

```



```

        Std = SQRT(Variance)
        SET FILTER TO
RETURN
PROCEDURE Mzulu
PARAMETER XDATE
        STORE "01/01/" + substr("&XDATE",7,2) to Z1day
        STORE (CTOD("&XDATE") - CTOD(Z1day)+1) TO Z2DAY
        STORE SUBSTR("&XDATE",8)+SUBSTR(SIR(1000000+Z2day),8,3) to MZULU
RETURN
*-----* EOF ANPROC.Procedure*-----*

```

5. MANAGEFL

a. MANAGEFL MENU

```
*****
*****      MANAGEFL.PRG      *****
*****
* Program.: Managefl.PRG      *
* Author...: PARK, TAEYONG    *
* Date.....: AUG 21, 1987     *
* Notice...: Copyright 1987   *
* Notes....: Menu file for file management system, called by pmain.prg *
*****
```

```
Do while .T.
  check = "X"
  CLEAR
  @ 1,15 TO 3,55 double
  @ 4,1 TO 23,77
  @ 6,3 TO 21,37
  @ 5,4 TO 7,20
  @ 6,5 SAY SPACE(15)
  @ 6,41 TO 12,75
  @ 5,42 TO 7,59
  @ 6,43 SAY SPACE(16)
  @ 14,41 TO 21,75
  @ 13,42 TO 15,59
  @ 14,43 SAY SPACE(16)
  @ 2,17 SAY "M a n a g e m e n t   o f   f i l e s"
  @ 6,6 SAY "  Edit files"
  @ 6,44 SAY "  Addition"
  @ 14,44 SAY "Information"
  @ 8,47 SAY "1.To add Customer (CI)"
  @ 9,47 SAY "2.To add Property (SN)"
  @ 10,47 SAY "3.To add ASL (SN)"
  @ 9,7 SAY "4.To change Customer (CI)"
  @ 11,7 SAY "5.To change Property (SN)"
  @ 13,7 SAY "6.To change ASL (SN)"
  @ 15,7 SAY "7.To change Master (SN+REQNO)"
  @ 17,7 SAY "8.To change Batch (SN+REQNO)"
  @ 20,7 SAY "9.Query on files"
  @ 16,47 SAY "Today is ....."
  @ 16,60 SAY today
  @ 18,43 SAY "Stock number : "
  @ 19,55 SAY "(Or/And)"
  @ 20,43 SAY "Customer code : "
  SET COLOR TO N/W
  @ 18,59 SAY Stockn pict "9999-99-999-9999"
  @ 20,59 SAY mci pict "9999"
  SET COLOR TO
  STORE "" TO SEL
  @ 22,58 SAY SEL
  @ 22, 8 SAY "Enter Selection (1 - 9, or 0 to go to mainmenu) : :|"
  *.....The following lines are for select loop
  i=0
  DO WHILE i=0
    i=INKEY()
    @ 22,58 SAY ""
    IF UPPER(CHR(i))$"0123456789"
      @ 22,58 SAY CHR(i)
      EXIT
    ENDIF
    i=0
  ENDDO
  Do case
    Case CHR(i) = '1'
      @ 18,59 SAY "XXXX-XX-XXX-XXXX"
      @ 20,59 get mci pict "9999"
      READ
```

```

DO ADDCUST
Case CHR(I) = '2'
  @ 18,59 get stockn pict "9999-99-999-9999"
  @ 20,59 SAY "XXXX"
  READ
DO ADDPROP
Case CHR(I) = '3'
  @ 18,59 get stockn pict "9999-99-999-9999"
  @ 20,59 Say "XXXX"
  READ
DO ADDASL
Case CHR(I) = '4'
  @ 18,59 SAY "XXXXXX-XX-XXX-XXXX"
  @ 20,59 get mci pict "9999"
  READ
DO EDITCUST
Case CHR(I) = '5'
  @ 18,59 get stockn pict "9999-99-999-9999"
  @ 20,59 Say "XXXX"
  READ
DO EDITPROP
Case CHR(I) = '6'
  @ 18,59 get stockn pict "9999-99-999-9999"
  @ 20,59 Say "XXXX"
  READ
DO EDITASL
Case CHR(I) = '7'
  @ 18,42 clear to 20, 65
  @ 18,43 SAY "Stock number : " get stockn pict;
  "9999-99-999-9999"
  @ 19,43 Say "Customer code : " get mci pict "9999"
  @ 20,43 Say "Transaction type : " get mtype pict "!!"
  READ
DO EDITMAST
Case CHR(I) = '8'
  @ 18,42 clear to 20, 65
  @ 18,43 SAY "Stock number : " get stockn pict;
  "9999-99-999-9999"
  @ 19,43 Say "Customer code : " get mci pict "9999"
  @ 20,43 Say "Transaction type : " get mtype pict "!!"
  READ
DO EDITBAT
Case CHR(I) = '9'
  Do Manageq
  CHECK = "I"
CASE I=27 .OR. CHR(I) = '0'
  ? Chr(7)
  RETURN
OTHERWISE
  ? CHR(7)
ENDCASE
ENDDO
CLOSE DATABASES
RETURN
*-----* Eof: Managefl.prg *-----*

```

b. ADDASL

```

*****
***** ADDASL.PRG *****
*****
* Module name.....: ADDASL.prg *
* Author.....: Park,Taeyong *
* Date.....: Aug 25. 1987 *
* Purpose.....: Add Authorized Storage List item(ASL) into *
* ASL file by manual *
* Called by.....: MANAGEFL.prg *

```

```

* Modules called : None
* Variables used.:
* Public.: CHECK.
*****
If stockn = " "
    Return
Endif
Select A
Use ASL INDEX INasl
Select B
Use property INDEX Snproper
Select A
*----- Set up loop for adding new ASL.
Store "X" to dcheck, fcheck
Dloop = .T.
Do while Dloop
    Clear
    *----- check to see if stock number already exists.
    Seek StockN
    Do case
        Case found()
            @ 15,10 say SN + " is Already exist in ASL file !"
            @ 17, 20 say " Do you want to change ? " get check
            Read
            Check =upper (check)
            If check = "Y"
                Clear
                Set format to Editasl
                Read
                Set format to
            Endif
        Case .not. found()
            Clear
            Append blank
            Replace SN with StockN
            Replace date with date()
            Do while .T.
                Set format to ASL
                Read
                If Upper(dcheck) = "Y"
                    Set format to
                    Exit
                Endif
            Enddo
        Endcase
    Store "X" to tcheck
    Clear
    @ 17,20 say " Do you want to check property file?" get tcheck
    Read
    Tcheck = upper(tcheck)
    If Tcheck = "Y"
        Select B
        Clear
        Seek StockN
        Do case
            Case found()
                Store "X" to tcheck
                @ 15,15 say StockN + " is already exist in property file "
                @ 17,20 say " Do you want to change ? " get tcheck
                Read
                Tcheck = upper(tcheck)
                If tcheck = "Y"
                    Clear
                    Set format to Editprop
                    Edit recno()
                    Set format to
                Endif
                Dloop = .F.
            Case .NOT. found()

```

```

        Clear
        Append blank
        Replace SN with StockN
        Replace date with date()
        Do while .T.
            Set format to prop
            Read
            fcheck = upper(fcheck)
            If fcheck = "Y"
                Replace tvalue with unitcost * onhand
                Set format to
                Dloop = .F.
            Exit
        Endif
    Enddo
Endcase
Endif
Dloop = .F.
Enddo
Close all
Store space(16) to stockn
Release all
Return
*-----* EOF Addas1.prg *-----*

```

c. ADDPROP

```

*****
*****                      ADDPROP.PRG                      *****
*****
* Module name.....: ADDPROP.prg                               *
* Author.....: Park, Taeyong                                   *
* Date.....: Aug 20. 1987                                     *
* Purpose.....: Add new properties into property file         *
* Called by.....: MANAGEFL.prg                                 *
* Modules called : None                                         *
* Variables used..:                                           *
*      Public.: TODAY                                           *
*****
If StockN = " "
    Return
Endif
Clear
Select A
Use ASL INDEX INas1
Select B
Use property INDEX Snproper
Today = date()
check = "X"
stock = .T.
Do while Stock
    clear
    Seek StockN
    Store "X" to fcheck
    Do case
        Case found()
            clear
            Store "X" to tcheck
            @ 15,10 say StockN + " is already exist in property file"
            @ 17, 20 Say "Do you want to change it NOW ?" get tcheck
            Read
            Tcheck = upper(tcheck)
            If tcheck = "N"
                clear
                Stock = .F.
            Exit

```

```

        Endif
        Set format to Editprop
        Edit recno()
        Replace tvalue with unitcost * onhand
        Set format to
        Exit
    Case .NOT. found()
        Append blank
        Replace SN with StockN
        Replace date with date()
        Do while .T.
            Set format to prop
            Read
            fcheck = upper(fcheck)
            If fcheck = "Y"
                Replace tvalue with unitcost * onhand
                Set format to
                Stock = .F.
            Exit
        Endif
    Enddo
Endcase
clear
Store "X" to acheck
@ 15, 20 SAY "Is this " + stockN + " ASL item ?" get acheck
Read
Acheck = upper (acheck)
If acheck = "Y"
    Select A
    Seek stockN
    Store "X" to dcheck
    Do case
        Case found()
            @ 20,10 say SN + " is Already exist !,;
            Do you want to change ? " get check
            Read
            check =upper (check)
            If check = "Y"
                Set format to EDITASL
                Edit Recno()
                Set format to
                Exit
            Endif
        Case .not. found()
            Append blank
            Replace SN with StockN
            Replace date with date()
            Do while .T.
                Set format to ASL
                Read
                Dcheck= upper(dcheck)
                If dcheck = "Y"
                    Set format to
                    Exit
                Endif
            Enddo
        Endcase
    Endif
Enddo
Close all
Store space(16) to stockn
Release all
Return
*-----* Eof Addprop.prg *-----*

```


d. ADDCUST

```

*****
*****          ADDCUST.PRG          *****
*****
* Module name.....: ADDCUST.prg      *
* Author.....: Park, Taeyong         *
* Date.....: Aug 25. 1987            *
* Purpose.....: Add customer into customer file      *
* Called by.....: MANAGEFL.prg       *
* Modules called : None              *
*****

Use Customer index Cicust
Cust= .T.
Store "Y" to Dcheck
Do while Cust
    *----- check proposed customer code
    Seek Mci
    Do case
        Case Mci = "      "
            Clear
            Cust = .F.
        Case found()
            @ 10,15 clear to 15,65
            @ 10,15 to 15,65
            @ 12,20 say "Customer code " + Mci + " Already exists !"
            @ 13,27 Say "Press any key to return....."
            ? chr(7)
            Mci = space(4)
            Wait " "
        Case .not. found()
            Append blank
            Replace CI with mci
            Replace date with date()
            Do while .T.
                Set format to cust
                Read
                If dcheck = "Y"
                    Set format to
                    Replace fundoh with fund - expend
                    Mci = space(4)
                    Exit
                Else
                    Chr(7)
            Endif
        Enddo
    Endcase
Enddo
Store mtype to mtypel
If mtypel <> "  "
    Return
Endif
Close all
Release all
Return
*-----* EOF addcust.prg *-----*

```

e. EDITASL

```

*****
*****          EDITASL.PRG          *****
*****
* Module name.....: EDITASL.prg      *
* Author.....: Park, Taeyong         *
* Date.....: Aug 28. 1987            *
* Purpose.....: Edit(delete or change) contents of entry      *

```

```

*                               of ASL file.                               *
* Called by.....: MANAGEFL.prg                                         *
* Modules called : ASL.fmt                                              *
*****
Use ASL index INASL
Clear
More = .T.
Do while More
  Seek STOCKN
  Do case
    Case STOCKN = " "
      More = .F.
    Case found()
      Set format to EDITASL
      Edit recno()
      Set format to
      Store "Y" to check
      @ 15, 20 Say "Pack marked records NOW ? (Y/N)" get check pict "!"
      Read
      If check = "Y"
        Set talk on
        Pack
        Set talk off
        Stockn = space(16)
        more = .F.
      Endif
    Case .NOT. found()
      @ 15,30 Say "No such STOCK NUMBER code in the file !"
      More = .F.
      Stockn = space(16)
      Wait
  Endcase
Enddo (while more)
Close all
Release all
Return
*-----* Eof EDITPROP.prg *-----*

```

f. EDITBAT

```

*****
***** DEITBAT.PRGM *****
*****
* Module name.....: EDITBAT.prg                                         *
* Author.....: Park, Taeyong                                           *
* Date.....: Aug 28. 1987                                              *
* Purpose.....: Edit Batch file                                       *
* Called by.....: MANAGEFL.prg                                         *
* Modules called : EDITBAT.fmt                                          *
* Variables used..:                                                    *
*   Public.: STOCKN,TODAY,MVNC                                         *
*   Local...:                                                         *
*****
Use Batch index SNCIBAT
Clear
More = .T.
Do while More
  Store space(16) to MVNC
  @ 15,10 Say "Enter Voucher number if you know : " get mvnc
  Read
  Do case
    Case STOCKN = " " .AND. Mvnc = " "
      Clear
      Close all

```

```

        Return
    Case stockn # " " .AND. Mvnc = " "
        Search = stockn
    Case stockn # " " .AND. Mvnc # " "
        Search = stockn + upper(mvnc)
    Case stockn = " " .AND. Mvnc # " "
        Search = upper(Mvnc)
Endcase
Seek search
If found()
    Set format to EDITBAT
    Edit recno()
    Set format to
    Store "Y" to check
    @ 15, 20 Say "Pack marked records NOW ? (Y/N)" get check pict "!"
    Read
    If check = "Y"
        Set talk on
        Pack
        Set talk off
        Stockn = space(16)
        more = .F.
    Endif
Else
    @ 15,10 Say "No such STOCK NUMBER or Voucher Number in the file !"
    More = .F.
    Stockn = space(16)
    Wait
Endif
Enddo
Close databases
Release all
Return
*-----* Eof Editbat.prg *-----*

```

g. EDITCUST

```

*****
***** EDITCUST.PRG *****
*****
* Module name....: EDITCUST.prg *
* Author.....: Park, Taeyong *
* Date.....: Aug 28. 1987 *
* Purpose.....: Edit Customer record from customer file *
* Called by.....: MANAGEFL.prg *
* Modules called : EDITCUST.fmt *
* Variables used... *
* Local...: MCI *
*****
Use CUSTOMER index CICUST
Clear
More = .T.
Do while More
    Seek Mci
    Do case
        Case mci = " "
            More = .F.

        Case found()
            Set format to EDITCUST
            Edit recno()
            Set format to

            Store "Y" to check
            @ 15, 20 Say "Pack marked records NOW ? (Y/N)" get check pict "!"

```

```

Read
  If check = "Y"
    Set talk on
    Pack
    Set talk off
    Mci = space(4)
    more = .F.
  Endif
Case .NOT. found()
  @ 15,35 Say "No such customer code in file !"
  More = .F.
  Mci = space(4)
  Wait
Endcase
Enddo (while more)
Close all
Release all
Return
*-----* Eof EDITcust.prg *-----*

```

h. EDITMAST

```

*****
***** EDITMAST.PRG *****
*****
* Module name....: EDITMAST.prg *
* Author.....: Park, Taeyong *
* Date.....: Aug 28. 1987 *
* Purpose.....: Edit master file *
* Called by.....: MANAGEFL.prg *
* Modules called : EDITMAST.fmt *
* Variables used..: *
* Local...: STOCKN,MCI *
*****

```

Use MASTER index SCTMAST

```

Clear
@ 10,10 to 15,62
@ 11,30 Say "Warning!"
@ 13,15 Say "This process is not allowed to every person,"
@ 14,15 Say " Enter password to continue." get mpass pict "!!!!"
Read
If pass <> mpass
  Close all
  Return
Endif
Set filter to SN=stockn .AND. ci=mci
Clear
@ 10,10 Say "Enter the transaction type : " get mtype Pict "!!"
Read
More = .T.
Do while More
  Seek Mtype
  Do case
    Case STOCKN = " "
      More = .F.
    Case found()
      Set format to EDITMAST
      Edit recno()
      Set format to
      Store "Y" to check
      @ 15, 20 Say "Pack marked records NOW ? (Y/N)" get check pict "!"
      Read
      If check = "Y"
        Set talk on

```

```

        Pack
        Set talk off
        Mci = space(4)
        Stockn = space(16)
        more = .F.
    Endif
Case .NOT. found()
    @ 15,35 Say "No such STOCK NUMBER code in file !"
    More = .F.
    Stockn = space(16)
    Wait
Endcase
Enddo (while more)
Set filter to
Close all
Release all
Return
*-----* Eof EDITMAST.prg *-----*

```

i. EDITPROP

```

*****
***** EDITPROP.PRG *****
*****
* Module name.....: EDITPROP.prg *
* Author.....: Park, Taeyong *
* Date.....: Aug 28. 1987 *
* Purpose.....: Edit property file *
* Called by.....: MANAGEFL.prg *
* Modules called : EDITPROP.fmt *
* Variables used.: *
*     Public.: *
*     Local.: STOCKN,MCI *
*****
Use PROPERTY index SNPROPER
Clear
More = .T.
Do while More
    Seek STOCKN
    Do case
        Case STOCKN = " "
            More = .F.
        Case found()
            Set format to EDITPROP
            Edit recno()
            Set format to
            Store "Y" to check
            @ 15, 20 Say "Pack marked records NOW ? (Y/N)" get check pict "!"
            Read
            If check = "Y"
                Set talk on
                Pack
                Set talk off
                Mci = space(4)
                STOCKN=SPACE(16)
                more = .F.
            Endif
        Case .NOT. found()
            @ 15,35 Say "No such STOCK NUMBER code in file !"
            More = .F.
            Mci = space(4)
            Wait
    Endcase
Endcase

```

```

Enddo (while more)
Close all
Release all
Return
*-----* Eof EDITPROP.prg *-----*

```

j. MANAGEQ

```

*****
*****      MANAGEQ.PRG      *****
*****
* Module name.....: MANAGEQ.PRG      *
* Author.....: Park, Taeyong      *
* Date.....: OCT 28, 1987      *
* Purpose.....: CONSULT CONTENTS OF FILES      *
* Called by.....: MANAGEFL.prg      *
* Modules called : SETUP.FRM      *
* Variables used...      *
*      Public.:      *
*      Local...: STOCKN.MCI      *
*****
Close databases
Clear
Set talk off
set echo off
Use property
Index on Class to Clssprop
Use Property index snproper,Clssprop
Select B
Use ASL index inasl
Select A
Set relation to SN into B
title = "Query on Stock number"
Mok="Y"
Store space(16) to stockn
Store "" to Mclass,Mconf
Do while .T.
    Clear
    @ 4,25 say title
    @ 8,15 Say "Options"
    @ 11,17 Say "1. Stock number"
    @ 12,17 Say "2. Class"
    @ 13,17 Say "3. All"
    @ 15,17 Say "4. Return to main menu"
    @ 17,15 Say "Enter Option: "
    @ 2,1 to 23,75
    Store "" to sel
    @ 17,29 Get sel
    i=0
    Do while i=0
        i=inkey()
        If chr(i)$"1234"
            Exit
        Endif
        i=0
    Enddo
    @ 17,29 Say Chr(i)
    If chr(i)="4"
        Exit
    Endif
    Do case
        Case chr(i)="1"
            @ 11,36 Say "Enter stock number " ;
            Get Stockn pict "9999-99-999-9999"
            Read
            If Stockn=" "
                ? Chr(7)

```



```

        Loop
    Endif
    Set filter to Sn='&Stockn'
    Set Order to 1
    Do setup
    Report form stockrpl
    Set filter to
Case chr(i)="2"
    @ 12,36 Say "Enter Class Number:" Get Mclass pict "9"
    @ 13,38 Say "0| for class 10"
    Read
    If Mclass=" "
        ? Chr(7)
        Loop
    Endif
    Set filter to Class='&Mclass'
    Set Order to 2
    Do setup
    Report form stockrpl
    Set filter to
Case chr(i)="3"
    @ 13,26 Say "<- Is this your select(Y/N):" Get Mconf pict "!"
    Read
    If Mconf="N"
        ? Chr(7)
        Loop
    Endif
    Set Order to 2
    Do setup
    Report form stockrpl
Otherwise
    Loop
Endcase
If Upper(Mok)="Y"
    Set console on
    Set print off
    Set Order to 1
Else
    @ 24,17 Say "Press any key to continue..."
    Wait""
Endif
Clear
Enddo
Close databases
Release all
Erase clssprop.ndx
Store space(16) to stockn
Clear
Return
*-----* Eof MANAGEQ.prg *-----*
```

APPENDIX F

SCREEN FORMAT

1. TRANSACT

a. BATCH

```

*****
***** Module name....: Batch.fmt *****
*****
@ 2, 11 SAY "Requeset for issue"
@ 3, 47 SAY "Today ....."
@ 3, 61 SAY today
@ 5, 8 SAY "Requested From : "
@ 5, 25 SAY CUSTOMER->CDESC
@ 6, 8 SAY " Send To : The 150 Infantry Division"
@ 8, 8 SAY "Type of transaction : (RD) Request No : "
@ 8, 53 GET MREQNO PICT "9999!!-9999-9999"
@ 9, 10 SAY "Request for Issue to Div ( Customer's )"
@ 12, 18 SAY "Stock Number : "
@ 12, 32 GET BATCH->SN pict "9999-99-999-9999"
@ 14, 18 SAY "Description : "
@ 14, 32 GET PROPERTY->NM
@ 16, 18 SAY " Unit : "
@ 16, 32 GET PROPERTY->UNIT
@ 16, 39 SAY "Price : "
@ 16, 47 GET PROPERTY->UNITCOST
@ 16, 56 SAY "$"
@ 18, 18 SAY " Quantity : "
@ 18, 32 GET BATCH->QTY
@ 23, 30 SAY "Is this record correct ? " get batchcheck pict "!"
@ 1, 5 TO 3, 40
@ 4, 4 TO 20, 70
@ 7, 5 TO 7, 69
@ 10, 5 TO 10, 69
@ 8, 38 TO 9, 38
*-----* Eof.batch.fmt *-----*

```

b. MAST

```

*****
***** Module name....: Mast.fmt *****
*****
@ 2, 9 SAY mtitle
@ 3, 43 SAY "Today is..."
@ 3, 56 SAY Today
@ 5, 7 SAY mhost + mcdesc1
@ 6, 7 SAY " "+mcust + mcdesc2
@ 8, 6 SAY "Type of transaction : Voucher No : "
@ 8, 48 GET VN1 PICTURE "9999!!-9999-9999"
@ 9, 6 SAY Mtypea
@ 9, 34 SAY "Request Number : "
@ 9, 50 GET MREQNO PICTURE "9999!!-9999-9999"
@ 12, 19 SAY "Stock Number : "
@ 12, 32 GET MASTER->SN PICTURE "9999-99-999-9999"
@ 14, 19 SAY "Description : "
@ 14, 32 GET PROPERTY->NM
@ 16, 34 SAY "Unit : "
@ 16, 41 GET PROPERTY->UNIT FUNCTION "!AAA" PICTURE "XXXX"
@ 18, 19 SAY "Quantity : "
@ 18, 32 GET MASTER->QTY
@ 18, 50 SAY "Reusable? : "

```

```

@ 18, 62 GET MASTER->Misc
@ 20, 19 SAY "Price : "
@ 20, 32 GET PROPERTY->UNITCOST
@ 23, 30 SAY "Is this record correct ? " get zcheck pict "X"
@ 4, 4 TO 21, 71
@ 1, 6 TO 3, 40
@ 7, 5 TO 7, 70
@ 10, 5 TO 10, 70
@ 8, 32 TO 9, 32
*-----* Eof Mast.fmt *-----*

```

c. TURNIN

```

*****
***** Screen format for Turnin program *****
*****
@ 2, 11 SAY "Request for turn-in"
@ 3, 46 SAY "Today....."
@ 3, 58 SAY Today
@ 5, 6 SAY "From : "
@ 5, 13 SAY MCDESC1
@ 6, 8 SAY "To : "
@ 6, 13 SAY MCDESC2
@ 8, 6 SAY "Type of transaction : (TD) "
@ 9, 10 SAY "Request for turn-in Request No : "
@ 9, 52 GET BATCH->REQNO PICTURE "9999!!-9999-9999"
@ 11, 21 SAY "Stock number : "
@ 11, 37 GET BATCH->SN PICTURE "9999-99-999-9999"
@ 13, 21 SAY "Description : "
@ 13, 37 GET PROPERTY->NM
@ 15, 16 SAY "Unit : "
@ 15, 25 GET PROPERTY->UNIT PICTURE "!XXX"
@ 15, 37 SAY "Reuseable? : "
@ 15, 51 GET BATCH->MISC
@ 17, 16 SAY "Price : "
@ 17, 25 GET PROPERTY->UNITCOST
@ 17, 37 SAY "Quantity : "
@ 17, 51 GET BATCH->QTY
@ 20, 34 SAY "Is this record correct ?" Get Zcheck pict "!"
@ 1, 5 TO 3, 40
@ 4, 4 TO 19, 70
@ 7, 5 TO 7, 69
@ 4, 36 TO 4, 36
@ 4, 37 TO 4, 37
@ 10, 5 TO 10, 69
@ 8, 37 TO 9, 37
*-----* Eof TURNIN.fmt *-----*

```

d. CANCMAST

```

*****
***** Module name.....: CANCMAST.FMT *****
*****
@ 2, 9 SAY "Cancel Request for issue item"
@ 3, 44 SAY "Today .."+Dtoc(date())+"("+Zulu+")"
@ 6, 25 SAY "Stock number : "
@ 6, 40 SAY MASTER->SN
@ 8, 25 SAY "Customer Code : "
@ 8, 40 SAY MASTER->CI
@ 10, 24 SAY "Request number : "
@ 10, 40 SAY MASTER->REQNO
@ 13, 13 SAY "Quantity : "
@ 13, 24 SAY MASTER->QTY
@ 13, 33 SAY "Price : "
@ 13, 41 SAY MASTER->UNITCOST

```

```

@ 13, 51 SAY "Date :("
@ 13, 58 SAY MASTER->DATE
@ 13, 62 SAY ")"
@ 17, 14 SAY "Is this record what you want to cancel?(Y/N)" ;
Get checks pict "!"
@ 1, 6 TO 3, 40
@ 4, 5 TO 15, 70
@ 16, 5 TO 18, 70
*-----* Eof CANCMAS.T.fmt *-----*

```

2. MANAGEFL

a. ASL

```

*****
***** Module name....: Asl.fmt *****
*****
@ 2, 8 SAY "Authorized storage list file"
@ 3, 45 SAY "Today is ....." get today
@ 6, 20 SAY "Stock number"
@ 6, 44 GET Stockn PICTURE "9999-99-999-9999"
@ 9, 20 SAY "Reorder Point"
@ 9, 44 GET ASL->ROP
@ 11, 20 SAY "Safety Level"
@ 11, 44 GET ASL->SL
@ 13, 20 SAY "Requisition Objective :"
@ 13, 44 GET ASL->RO
@ 16, 20 SAY "Order Shipping Time :"
@ 16, 44 GET ASL->OST
@ 16, 50 SAY "days"
@ 18, 18 SAY "Resource control number :"
@ 18, 44 GET ASL->RCN PICTURE "9999"
@ 23, 35 SAY " Is this record correct ? : " get dcheck pict "!"
@ 1, 5 TO 3, 40
@ 4, 4 TO 20, 70
*-----* Eof.As1.fmt *-----*

```

b. CUST

```

*****
***** Cust.fmt (CUSTOMER file screen format called by ADDCUST)*****
*****
@ 2, 13 SAY "Customer File"
@ 3, 43 SAY "Today is ....."
@ 3, 62 SAY CUSTOMER->DATE
@ 6, 13 SAY "Customer Code"
@ 6, 28 SAY CUSTOMER->CI
@ 6, 47 SAY "Priority"
@ 6, 57 GET CUSTOMER->PRIORITY
@ 8, 22 SAY "Name"
@ 8, 28 GET CUSTOMER->CDESC
@ 10, 19 SAY "Address"
@ 10, 28 GET CUSTOMER->ADDRESS FUNCTION "S30";
PICTURE "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
@ 12, 28 GET CUSTOMER->ZIPCODE
@ 12, 37 SAY "Zip code"
@ 15, 18 SAY "Fund :"
@ 15, 28 GET CUSTOMER->FUND PICTURE "9999999.99"
@ 15, 40 SAY "$ Allowed"
@ 17, 28 GET CUSTOMER->EXPEND PICTURE "9999999.99"
@ 17, 40 SAY "$ Expenditure"
@ 18, 22 SAY "-----"

```

```
@ 23, 35 SAY "Is this record correct ? " get dcheck pict "!"
@ 1, 6 TO 3, 40
@ 4, 5 TO 20, 70
*-----* Eof CUST.fmt *-----*
```

c. PROP

```
*****
***** Module name....: Prop.fmt *****
*****
@ 2, 12 SAY "Property Book file"
@ 3, 44 SAY "Today ....."
@ 3, 62 SAY Today
@ 6, 22 SAY "Stock number :"
@ 6, 38 GET Stockn PICTURE "9999-99-999-9999"
@ 7, 22 SAY "Serial NO :"
@ 7, 38 GET PROPERTY->SERIALNO
@ 8, 38 GET PROPERTY->NM
@ 8, 22 SAY "Description :"
@ 10, 25 SAY "Unit :"
@ 10, 38 GET PROPERTY->UNIT
@ 10, 46 SAY "Class :"
@ 10, 54 GET PROPERTY->CLASS PICTURE "99"
@ 13, 25 SAY "On hand :"
@ 13, 38 GET PROPERTY->ONHAND
@ 15, 25 SAY "Price :"
@ 15, 38 GET PROPERTY->UNITCOST
@ 15, 47 SAY "$"
@ 16, 25 SAY "-----"
@ 18, 20 SAY "Is this combat essential item ? :"
@ 18, 54 GET PROPERTY->ESSENCE
@ 23, 35 SAY "Is this record correct ? : " get fcheck pict "!"
@ 1, 5 TO 3, 40
@ 4, 4 TO 20, 70
*-----* Eof. Prop.fmt *-----*
```

d. EDITBAT

```
*****
***** Editbat.fmt (BATCH file screen format called by EDITBAT) *****
*****
@ 2, 7 SAY "Edit(change or delete) Batch file"
@ 3, 46 SAY "Today....."
@ 3, 61 GET BATCH->DATE
@ 5, 23 SAY "Stock number :"
@ 5, 40 GET BATCH->SN
@ 7, 23 SAY "Voucher number :"
@ 7, 40 GET BATCH->REQNO
@ 9, 23 SAY "Customer Code :"
@ 9, 40 GET BATCH->CI
@ 11, 14 SAY "Type of action :"
@ 11, 35 GET BATCH->TYPE
@ 11, 50 SAY "Posted :"
@ 11, 60 GET BATCH->POSTED
@ 13, 14 SAY "Quantity :"
@ 13, 35 GET BATCH->QTY
@ 13, 50 SAY "Unitcost :"
@ 13, 60 GET BATCH->UNITCOST
@ 13, 69 SAY "$"
@ 14, 35 SAY "Date : "+Dtoc(BATCH->DATE)
@ 17, 7 SAY "(Insert mode) : Ins (Record) Next : PgDn"
@ 18, 7 SAY "(Delete) Character :Del Previous : PgUp"
@ 19, 13 SAY "Field: ctrl+Y Record:ctrl+ U (Done/Save) :;
End Abandon :Esc"
@ 1, 5 TO 3, 40
```



```
@ 4, 5 TO 15, 70
@ 16, 5 TO 20, 70
@ 17, 35 TO 19, 35
*-----* Eof EDITBAT.fmt *-----*
```

e. EDITCUST

```
*****
***** Editmast.fmt (MASTER file screen format called by EDITMAST)*****
*****
@ 2, 10 SAY "Edit/Change Master file"
@ 3, 47 SAY "Today....."
@ 3, 61 SAY DIOC{DATE()}
@ 5, 20 SAY "Stock Number :"
@ 5, 36 GET MASTER->SN PICTURE "9999-99-999-9999"
@ 7, 14 SAY "Customer code :"
@ 7, 31 GET MASTER->CI
@ 7, 43 SAY "Action Type :"
@ 7, 58 GET MASTER->TYPE
@ 9, 13 SAY "Request Number :"
@ 9, 31 GET MASTER->VN PICTURE "9999!!-9999-9999"
@ 10, 31 GET MASTER->REQNO PICTURE "9999!!-9999-9999"
@ 10, 50 SAY "Customer's"
@ 12, 12 SAY "Quantity :"
@ 12, 24 GET MASTER->QTY
@ 13, 12 SAY " Price :"
@ 13, 24 GET MASTER->UNITCOST
@ 13, 30 SAY "S Reusable ? :"
@ 13, 58 GET MASTER->MISC
@ 14, 29 SAY "Date :"
@ 14, 36 GET MASTER->DATE
@ 17, 7 SAY "(Insert mode) : Ins (Record) Next : PgDn"
@ 18, 7 SAY "(Delete) Character :Del Previous : PgUp"
@ 19, 13 SAY "Field:ctrl+Y Record: ctrl+U (Done/Save) :;
ctrl+End Abandon :Esc"
@ 1, 5 TO 3, 40
@ 4, 5 TO 15, 70
@ 16, 5 TO 20, 70
@ 17, 35 TO 19, 35
*-----* Eof EDITMAST.fmt *-----*
```

f. EDITASL

```
*****
***** Editasl.fmt (ASL screen format called by EDITASL)*****
*****
@ 2, 6 SAY "Edit Authorized Storage List file"
@ 3, 47 SAY "Today ....."
@ 3, 60 SAY DIOC{DATE()}
@ 5, 20 SAY "Stock number :"
@ 5, 36 GET ASL->SN
@ 8, 10 SAY "ReOrder Point :"
@ 8, 26 GET ASL->ROP
@ 8, 38 SAY "Order shipping time :"
@ 8, 60 GET ASL->OST
@ 8, 64 SAY "days"
@ 10, 11 SAY "Safety Level :"
@ 10, 26 GET ASL->SL
@ 10, 38 SAY " Date :"
@ 10, 60 GET ASL->DATE
@ 12, 10 SAY "Requisition Objective :"
@ 12, 34 GET ASL->RO
@ 14, 23 SAY "Resource control number :"
@ 14, 50 GET ASL->RCN
@ 17, 7 SAY "(Insert mode) : Ins (Record) Next : PgDn"
```



```

@ 18, 7 SAY "(Delete) Character :Del Previous : PgUp"
@ 19, 13 SAY "Field: Y Record: U (Done/Save) : End Abandon :E
@ 1, 5 TO 3, 40
@ 4, 5 TO 15, 70
@ 16, 5 TO 20, 70
@ 17, 35 TO 19, 35
*-----* Eof EDITASL.fmt *-----*

```

g. EDITPROP

```

*****
*** Editprop.fmt (PROPERTY file screen format called by EDITPROP)*****
*****
@ 2, 12 SAY "Edit property file"
@ 3, 46 SAY "Today ....."
@ 3, 61 SAY DTOC(TODAY)
@ 5, 7 SAY "Stock Number :"
@ 5, 23 GET PROPERTY->SN
@ 5, 45 SAY "Serial No :"
@ 5, 57 GET PROPERTY->SERIALNO
@ 7, 7 SAY "Nomenclature :"
@ 7, 23 GET PROPERTY->NM
@ 9, 15 SAY "Unit :"
@ 9, 23 GET PROPERTY->UNIT
@ 9, 30 SAY "Essential item? :"
@ 9, 48 GET PROPERTY->ESSENCE
@ 9, 54 SAY "Class :"
@ 9, 62 GET PROPERTY->CLASS
@ 12, 14 SAY "Onhand :"
@ 12, 23 GET PROPERTY->ONHAND
@ 12, 35 SAY "Price :"
@ 12, 49 GET PROPERTY->UNITCOST
@ 12, 59 SAY "$"
@ 14, 30 SAY "Total Value :"
@ 14, 49 GET PROPERTY->TVALUE
@ 14, 59 SAY "$"
@ 17, 7 SAY "(Insert mode) : Ins (Record) Next : PgDn"
@ 18, 7 SAY "(Delete) Character :Del Previous : PgUp"
@ 19, 13 SAY "Field: ctrl+Y Record: ctrl+U (Done/Save) : ;
ctrl+End Abandon :Esc"
@ 1, 5 TO 3, 40
@ 4, 5 TO 15, 70
@ 16, 5 TO 20, 70
@ 17, 35 TO 19, 35
*-----* Eof EDITPROP.fmt *-----*

```

h. EDITCUST

```

*****
***** Editcust.fmt (CUSTOMER file screen called by EDITCUST)*****
*****
@ 2, 9 SAY "Edit or Delete CUSTOMER file"
@ 3, 46 SAY "Today ...."
@ 3, 60 SAY TODAY
@ 5, 9 SAY "Customer :"
@ 5, 20 GET CUSTOMER->CI
@ 5, 25 SAY "code Priority :"
@ 5, 48 GET CUSTOMER->PRIORITY
@ 7, 20 GET CUSTOMER->CDESC
@ 7, 51 SAY "Description"
@ 8, 20 GET CUSTOMER->ADDRESS FUNCTION "S30";
PICTURE "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
@ 8, 51 SAY "Address"
@ 9, 20 GET CUSTOMER->ZIPCODE
@ 9, 51 SAY "Zipcode"

```

```

@ 11, 9 SAY "Fund allwed : "
@ 11, 23 GET CUSTOMER->FUND
@ 12, 23 GET CUSTOMER->EXPEND
@ 12, 36 SAY "Expend"
@ 13, 23 GET CUSTOMER->FUNDON
@ 13, 36 SAY "On hand"
@ 14, 33 SAY "Last edit date : "
@ 14, 51 GET CUSTOMER->DATE
@ 17, 7 SAY "(Insert mode) : Ins (Record) Next : PgDn"
@ 18, 7 SAY "(Delete) Character : Del Previous : PgUp"
@ 19, 13 SAY "Field:ctrl+ Y Record:ctrl+ U (Done/Save) : ;
ctrl+End Abandon :Esc"
@ 1, 5 TO 3, 40
@ 4, 5 TO 15, 70
@ 16, 5 TO 20, 70
@ 17, 35 TO 19, 35
*-----* Eof EDITCUST.fmt *-----*

```

3. ANALYSIS

a. EOQ

```

*****
***** Module name....: EOQ.fmt *****
*****
Clear
@ 2, 17 SAY "I N F O R M A T I O N  f o r  A N A L Y S I S"
@ 5, 12 SAY "You selected stock number : "
@ 5, 40 SAY PROPERTY->SN
@ 6, 17 SAY "("+Rtrim(PROPERTY->NM)+", "+"Unit:";
+Rtrim(PROPERTY->UNIT)+", Class:"+PROPERTY->CLASS+")"
@ 8, 18 SAY "Purchasing price(P) : "+Ltrim(str(property->unitcost));
+" $/"+Ltrim(PROPERTY->UNIT)
@ 10, 21 SAY "Annual Demand(R) : "+Ltrim(str(Rdem))+ " ";
+Rtrim(property->unit)+"/year"
@ 12, 14 SAY "Lead time(OST) in Month : "+Ltrim(str(mLeadt));
+" Month(s)"
@ 14, 21 SAY "Ordering cost(C) : $/order"
@ 14, 40 GET Orderc pict "9999.99"
@ 15, 11 SAY "Holding cost unit per year : %"
@ 15, 40 Get Frate pict "99.99"
@ 16, 5 SAY "(Select one of these)"
@ 17, 14 SAY "Stockout cost(If Known) : $/unit"
@ 17, 40 GET Scost pict "9999.99"
@ 17, 55 Say "Select?" get CSC pict "!"
@ 18, 16 SAY "Service Level in year : %"
@ 18, 40 GET Servl pict "99.9999"
@ 18, 55 SAY "Select?" get CSL pict "!"
@ 21, 5 SAY ""+title+"| Reorder point : " +Mrop
@ 21, 45 SAY "Safety Level : "+ Msl
@ 22, 15 SAY "Requisition objective : "+ Mro
@ 22, 45 SAY "Lead Time : "+Most
@ 24, 5 SAY space(50-Len(Ctitle))+(Ctitle) get check pict "!"
@ 1, 13 TO 3, 61 DOUBLE
@ 4, 1 TO 19, 74 DOUBLE
@ 20, 1 TO 23, 74 DOUBLE
*-----* Eof EOQ.fmt *-----*

```

LIST OF REFERENCES

1. Korea Army College, *Logistics Management*, Jinhae, Korea, April 30 1984
2. Administrative Refence 09-17-1, *National Defense logistics Terminology*, ROK DoD, Seoul, Korea, January 1985.
3. Tersine, Richard J., *Principles of Inventory and Materials Management*, North-Holland, second edition, 52 Vanderbilt Ave New York, NY, 10017 1982
4. Gee, Manwon, *Management Information System for the Division Level*, Korean institution of Defense, June 1986
5. Lee, Heeyung, Master's Thesis *The Development of a Standard Database for Republic of Korea Army's Logistics Support System*, Naval Postgraduate School, Monterey, CA,93943 March 1885
6. Howe, D.R., *Data analysis for database design*, Thomson litho Ltd, East Kilbride, Scotland 1984
7. Kroenke, David, *DATABASE Processing: Fundamentals, Design, Implementation*, Science Research Association, Inc. A subsidiary of IBM, Second Edition, Chicago, Henley-on-Thames, Sydney, Toronto
8. Carl Townsend *Mastering dBASEIII PLUS a structured approach*. Sybex, 2021 Challenger drive #100, Alameda, CA 94501. 1986

INITIAL DISTRIBUTION LIST

	No. Copies
1. Defense Technical Information Center Cameron Station Alexandria, VA 22304-6145	2
2. Library, Code 0142 Naval Postgraduate School Monterey, CA 93943-5002	2
3. Library, P.O.Box 77 Gong Neung Dong, Dobong Goo Seoul 130-09, Republic of Korea	1
4. Adjunct instructor Y.K. Mortagy Code 54My Department of Administrative Sciences Naval Postgraduate School Monterey, CA 93943	1
5. Professor Thomas. Moore Code 54 Mo Department of Administrative Sciences Naval Postgraduate School Monterey, CA 93943	1
6. Kim, Nak-Heung 176-100 Pookahyun-Dong Sudaemoon-Koo Seoul, 120-00 Republic of Korea	1
7. Kim, Jae Soo #4-2, 375 Hwanghak-Dong Jung-Koo Seoul, 100-00 Republic of Korea	1
8. Kim, Changwhan SMC 2889, Naval Postgraduate School Monterey, CA 93943	1
9. Erdal Ozturk Gazipasa Mah. Zeytinlik Cad. No.5/15 Trabzon Turkey	1

10. Park, Taeyong 2
Kyungbuk kumrung-gun Umo-myun Okgye-dong 821
640-74
Republic of Korea
11. Son, Gun ha 1
Unpyung-gu Susaek-dong Kukdae-won (NDPS) Apt Ka-dong 306
Seoul 120-00
Republic of Korea

Thesis

P157586 Park

c.1

A relational database
management system for a
ROK Army Infantry Divi-
sion with probabilistic
inventory control model.



thesP157586

A relational database management system



3 2768 000 78556 2

DUDLEY KNOX LIBRARY C. 10